intel®

OEM Boards, Systems & Software 1994

Multibus I Products

Multibus II Products

OEM Microcomputer Platforms

iRMX® Real-time Software

UNIX Multibus II Software*

intel®

OEM Boards, Systems & Software



intel®

U.S. and CANADA LITERATURE ORDER FORM

NAME: _____

COMPANY: _____

ADDRESS: _____

CITY: _____ STATE: _____ ZIP: _____

COUNTRY: _____

PHONE NO.: ( ) _____

ORDER NO.	TITLE	QTY.		PRICE		TOTAL
			×		=	
			×		=	
			×		=	
			×		=	
			×		=	
			×		=	
			×		=	
			×		=	
			×		=	
			×		=	

Subtotal _____

Must Add Your Local Sales Tax _____

Include postage:
Must add 15% of Subtotal to cover U.S. and Canada postage. (20% all other.)

→ Postage _____

Total _____

Pay by check, money order, or include company purchase order with this form ($200 minimum). We also accept VISA, MasterCard or American Express. Make payment to Intel Literature Sales. Allow 2-3 weeks for delivery.

☐ VISA ☐ MasterCard ☐ American Express Expiration Date _____

Account No. _____

Signature _____

Mail To: Intel Literature Sales
P.O. Box 7641
Mt. Prospect, IL 60056-7641

International Customers outside the U.S. and Canada should use the International order form on the next page or contact their local Sales Office or Distributor.

For phone orders in the U.S. and Canada, call Toll Free: (800) 548-4725 or FAX to (708) 296-3699. Please print clearly in ink to expedite your order.

Prices good until 12/31/94.
Source HB

intel

INTERNATIONAL LITERATURE ORDER FORM

NAME: _____

COMPANY: _____

ADDRESS: _____

CITY: _____ STATE: _____ ZIP: _____

COUNTRY: _____

PHONE NO.: ( ) _____

ORDER NO.	TITLE	QTY.		PRICE	TOTAL
			×		=
			×		=
			×		=
			×		=
			×		=
			×		=
			×		=
			×		=
			×		=
			×		=

Subtotal _____

Must Add Your Local Sales Tax _____

Total _____

PAYMENT

Cheques should be made payable to your ***local*** Intel Sales Office (see inside back cover).

Other forms of payment may be available in your country. Please contact the Literature Coordinator at your ***local*** Intel Sales Office for details.

The completed form should be marked to the attention of the LITERATURE COORDINATOR and returned to your ***local*** Intel Sales Office.

intel®

OEM BOARDS, SYSTEMS, and SOFTWARE HANDBOOK

1994

Intel Corporation makes no warranty for the use of its products and assumes no responsibility for any errors which may appear in this document nor does it make a commitment to update the information contained herein.

Intel retains the right to make changes to these specifications at any time, without notice.

Contact your local Intel sales office or your distributor to obtain the latest specifications before placing your product order.

MDS is an ordering code only and is not used as a product name or trademark of Intel Corporation.

Intel Corporation and Intel's FASTPATH are not affiliated with Kinetics, a division of Excelan, Inc. or its FASTPATH trademark or products.

*Other brands and names are the property of their respective owners.

Additional copies of this document or other Intel literature may be obtained from:

Intel Corporation
Literature Sales
P.O. Box 7641
Mt. Prospect, IL 60056-7641

or call 1-800-879-4683

©INTEL CORPORATION, 1993

LGCPY1/100693

DATA SHEET DESIGNATIONS

Intel uses various data sheet markings to designate each phase of the document as it relates to the product. The marking appears in the upper, right-hand corner of the data sheet. The following is the definition of these markings:

Data Sheet Marking	Description
Product Preview	Contains information on products in the design phase of development. Do not finalize a design with this information. Revised information will be published when the product becomes available.
Advanced Information	Contains information on products being sampled or in the initial production phase of development.*
Preliminary	Contains preliminary information on new products in production.*
No Marking	Contains information on products in full production.*

*Specifications within these data sheets are subject to change without notice. Verify with your local Intel sales office that you have the latest data sheet before finalizing a design.



Multibus II
Single Board Computers 1

Multibus II
Memory Expansion Boards 2

Multibus II I/O Products and
MIX Boards 3

Multibus II
Development Systems
and Accessories 4

Multibus II
Architecture 5

iSBX Expansion Modules 6

Real-Time Software 7

Multibus I
Single Board Computers 8

Multibus I
Memory Expansion Boards 9

Multibus I
Peripheral Controllers 10

Multibus I
Serial Communication Boards 11

Multibus I
Digital and Analog
I/O Boards 12

Multibus I
System Packaging and
Power Supplies 13

Local Area Network
Boards and Software 14

Distributed Control Modules 15

Microcomputer Platforms 16

Support Products 17

Table of Contents

Alphanumeric Index xiv

MULTIBUS II PRODUCTS

CHAPTER 1

Multibus II Single Board Computers

iSBC 486/166SE and iSBC 486/133SE Single Board Computers 1-1
iSBC 486/150, 486/125 and iSBC 386/133 Single Board Computers 1-5
iSBC 386/120 Multibus II Single Board Computer 1-10
iSBC 486/DX66, iSBC 486/DX33 and 486/SX25 PC-Compatible Boards 1-18
UNIX OEM Developer Kit for Multibus II 1-23

CHAPTER 2

Multibus II Memory Expansion Boards

High Performance Memory Modules 2-1
iSBC MEM/128 Memory Board 2-5

CHAPTER 3

Multibus II I/O Products and MIX Boards

MIX Architecture 3-1
MIX Baseboards 3-4
MIX I/O Modules 3-7
I/O Boards 3-12
Multibus II Bus Interface Silicon Products 3-19
iSBC 387MX25 Math Extension Module 3-22
Firmware Development Package 3-23

CHAPTER 4

Multibus II Development Systems and Accessories

System 520 Chassis for Multibus II 4-1
System Enclosures & Backplanes 4-3
iSBC PKG/902, iSBC PKG/903 Multibus II iLBX II Backplanes 4-7
iSBC CSM/002 Central Services Module 4-10
iSBC CSM/001 Central Services Module 4-13

CHAPTER 5

Multibus II Architecture

The Multibus II Bus Structure 5-1

MULTIBUS II APPLICATION NOTES

AP-422 Designing a Central Services Module for Multibus II 5-17
AP-423 The Multibus II Interconnect Design Guide 5-46
AP-431 Multibus II Simplifies Partitioning of a Complex Design 5-125
AP-433 Simple I/O Design Example Using the Multibus II MIX Architecture 5-173
AP-437 Designing a Multibus II CPU Board with the Message Passing CoProcessor (MPC) 82389 5-221

iSBX BUS PRODUCTS

CHAPTER 6

iSBX Expansion Modules

iSBX GRAPHICS MODULES

iSBX 279A Display Subsystem 6-1

iSBX DIGITAL AND ANALOG I/O BOARDS

iSBX 311 Analog Input MULTIMODULE Board 6-5
iSBX 328 Analog Output MULTIMODULE Board 6-8
iSBX 350 Parallel I/O MULTIMODULE Board 6-11

Table of Contents (Continued)

iSBX 488 GPIB MULTIMODULE Board 6-14
iSBX SERIAL COMMUNICATION BOARDS
iSBX 351 Serial I/O MULTIMODULE Board 6-17
iSBX 354 Dual Channel Serial I/O MULTIMODULE Board 6-21

CHAPTER 7
Real-Time Software
REAL-TIME OPERATING SYSTEM SOFTWARE
iRMX for Windows Operating System 7-1
iRMX for PCs Operating System 7-5
iRMK Version I.3 Real-Time Kernel 7-9
iRMX III Real-Time Operating System 7-19
Intel System Release UNIX Software 4.03 7-23

MULTIBUS I PRODUCTS
CHAPTER 8
Multibus I Single Board Computers
iSBC 486/12DX2S, 486/12S and 386/12 Single Board Computers 8-1
iSBC 386 SX Single Board Computer and iSBC 272 VGA Graphics Module 8-6
iSBC 386/21/22/24/28 and 386/32/34/38 Single Board Computers 8-13
iSBC 286/12, 286/14, 286/16 Single Board Computers 8-21
iSBC 286/10A Single Board Computer 8-28
iSBC 186/03A Single Board Computer 8-33
iSBC86/35A, 86C/38, and 86/05A Single Board Computers 8-42
iSBC 88/25 Single Board Computer 8-47
iSBC 80/24A Single Board Computer 8-53
iSBC 80/10B Single Board Computer 8-59
iSBC 337A MULTIMODULE Numeric Data Processor 8-64

CHAPTER 9
Multibus I Memory Expansion Boards
DRAM EXPANSION BOARDS
iSBC 010EX, 020EX, and 040EX High Performance RAM Boards 9-1
iSBC 020CX iLBX RAM Boards 9-5
iSBC 314 512K Byte RAM MULTIMODULE Board 9-9
iSBC 304 128K Byte RAM MULTIMODULE Board 9-13
EPROM EXPANSION BOARDS
iSBC 429 Universal Site Memory Expansion Board 9-16
iSBC 428 Universal Site Memory Expansion Board 9-19
iSBC 341 28-Pin MULTIMODULE EPROM 9-23
See Chapter 2 for Multibus I Memory Modules

CHAPTER 10
Multibus I Peripheral Controllers
iSBC 221 Peripheral Controller 10-1
See Chapter 6 for iSBX Peripheral Controllers

CHAPTER 11
Multibus I Serial Communication Boards
iSBC 548/549 Terminal Controllers 11-1
iSBC 188/56 Advanced Communicating Computer 11-5
iSBC 544A Intelligent Communications Controller 11-14
iSBC 534 Four Channel Communication Expansion Board 11-20
iSBC 88/45 Advanced Data Communications Processor Board 11-24
See Chapter 6 for iSBX Serial Communication Boards

Table of Contents (Continued)

CHAPTER 12

Multibus I Digital and Analog I/O Boards

iSBC 519A Programmable I/O Expansion Board 12-1
iSBC 556 Optically Isolated I/O Board 12-3
iSBC 569 Intelligent Digital Controller 12-6
See Chapter 6 for iSBX Digital and Analog I/O Boards

CHAPTER 13

Multibus I System Packaging and Power Supplies

System 320 Chassis for Multibus I 13-1
iSBC 604/614 Modular Cardcage Assemblies 13-3
iSBC 608/618 Cardcages 13-6

LOCAL AREA NETWORK PRODUCTS

CHAPTER 14

Local Area Network Boards and Software

OpenNET Local Area Network Family 14-1
iSBX 586 Ethernet Data Link Engine 14-17
iSBC 552A Network Interface Adaptor 14-19
iSBC 186/51 Communicating Computer 14-22
See Chapter 3 for Multibus II Networking Board Products

BITBUS PRODUCTS

CHAPTER 15

Distributed Control Modules

BITBUS BOARDS

iSBX 344A BITBUS Intelligent MULTIMODULE Board 15-1
iRCB 44/10A BITBUS Digital I/O Remote Controller Board 15-9
iRCB 44/20A Analog I/O Controller 15-17

SYSTEMS PRODUCTS

CHAPTER 16

Microcomputer Platforms

MICROCOMPUTER PLATFORM SYSTEMS

Professional Workstations 16-1
Classic-E Platforms 16-7
Classic-R Platforms 16-11
Xpress Desktop Platforms 16-15
Xpress/MX Server Platforms 16-21
Xpress/LX Deskside Platforms 16-26
Xpress Disk Array Controller 16-32
Champion Low Profile Platform 16-34
Champion Mid-Tower Platform 16-37
Champion Expandable Desktop Platform 16-40

INDUSTRIAL MICROCOMPUTER SYSTEMS

OMNI Rack 16-43
XpressRack System 16-49

SERVICE AND SUPPORT

CHAPTER 17

Support Products

iRUG Description 17-1
Smart FIX Product Repair Services 17-2
Specialized Product Center 17-10
Intel Systems Engineering Services 17-13

Alphanumeric Index

AP-422 Designing a Central Services Module for Multibus II 5-17
AP-423 The Multibus II Interconnect Design Guide 5-46
AP-431 Multibus II Simplifies Partitioning of a Complex Design 5-125
AP-433 Simple I/O Design Example Using the Multibus II MIX Architecture 5-173
AP-437 Designing a Multibus II CPU Board with the Message Passing CoProcessor (MPC) 82389 5-221
Champion Expandable Desktop Platform 16-40
Champion Low Profile Platform 16-34
Champion Mid-Tower Platform 16-37
Classic-E Platforms 16-7
Classic-R Platforms 16-11
High Performance Memory Modules 2-1
Intel System Release UNIX Software 4.03 7-23
Intel Systems Engineering Services 17-13
iRCB 44/10A BITBUS Digital I/O Remote Controller Board 15-9
iRCB 44/20A Analog I/O Controller 15-17
iRMK Version I.3 Real-Time Kernel 7-9
iRMX for PCs Operating System 7-5
iRMX for Windows Operating System 7-1
iRMX III Real-Time Operating System 7-19
iRUG Description 17-1
iSBC 010EX, 020EX, and 040EX High Performance RAM Boards 9-1
iSBC 020CX iLBX RAM Boards 9-5
iSBC 186/03A Single Board Computer 8-33
iSBC 186/51 Communicating Computer 14-22
iSBC 188/56 Advanced Communicating Computer 11-5
iSBC 221 Peripheral Controller 10-1
iSBC 286/10A Single Board Computer 8-28
iSBC 286/12, 286/14, 286/16 Single Board Computers 8-21
iSBC 304 128K Byte RAM MULTIMODULE Board 9-13
iSBC 314 512K Byte RAM MULTIMODULE Board 9-9
iSBC 337A MULTIMODULE Numeric Data Processor 8-64
iSBC 341 28-Pin MULTIMODULE EPROM 9-23
iSBC 386 SX Single Board Computer and iSBC 272 VGA Graphics Module 8-6
iSBC 386/120 Multibus II Single Board Computer 1-10
iSBC 386/21/22/24/28 and 386/32/34/38 Single Board Computers 8-13
iSBC 428 Universal Site Memory Expansion Board 9-19
iSBC 429 Universal Site Memory Expansion Board 9-16
iSBC 486/12DX2S, 486/12S and 386/12 Single Board Computers 8-1
iSBC 486/150, 486/125 and iSBC 386/133 Single Board Computers 1-5
iSBC 486/166SE and iSBC 486/133SE Single Board Computers 1-1
iSBC 486/DX66, iSBC 486/DX33 and 486/SX25 PC-Compatible Boards 1-18
iSBC 519A Programmable I/O Expansion Board 12-1
iSBC 534 Four Channel Communication Expansion Board 11-20
iSBC 544A Intelligent Communications Controller 11-14
iSBC 548/549 Terminal Controllers 11-1
iSBC 552A Network Interface Adaptor 14-19
iSBC 556 Optically Isolated I/O Board 12-3
iSBC 569 Intelligent Digital Controller 12-6
iSBC 604/614 Modular Cardcage Assemblies 13-3
iSBC 608/618 Cardcages 13-6
iSBC 80/10B Single Board Computer 8-59
iSBC 80/24A Single Board Computer 8-53
iSBC 88/25 Single Board Computer 8-47

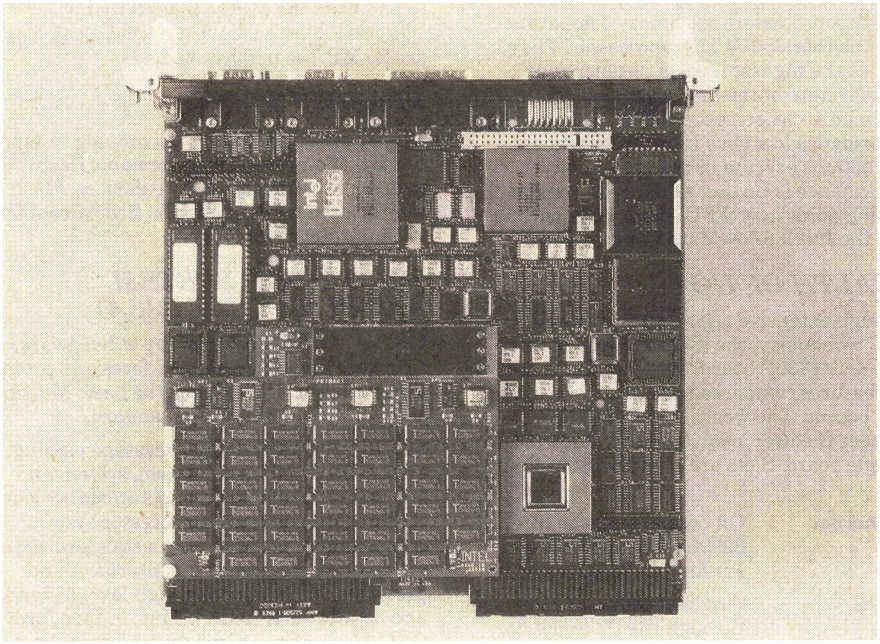
Alphanumeric Index (Continued)

iSBC 88/45 Advanced Data Communications Processor Board 11-24
iSBC CSM/001 Central Services Module 4-13
iSBC CSM/002 Central Services Module 4-10
iSBC MEM/128 Memory Board 2-5
iSBC PKG/902, iSBC PKG/903 Multibus II iLBX II Backplanes 4-7
iSBC86/35A, 86C/38, and 86/05A Single Board Computers 8-42
iSBX 279A Display Subsystem 6-1
iSBX 311 Analog Input MULTIMODULE Board 6-5
iSBX 328 Analog Output MULTIMODULE Board 6-8
iSBX 344A BITBUS Intelligent MULTIMODULE Board 15-1
iSBX 350 Parallel I/O MULTIMODULE Board 6-11
iSBX 351 Serial I/O MULTIMODULE Board 6-17
iSBX 354 Dual Channel Serial I/O MULTIMODULE Board 6-21
iSBX 488 GPIB MULTIMODULE Board 6-14
iSBX 586 Ethernet Data Link Engine 14-17
MIX Architecture 3-1
OMNI Rack 16-43
OpenNET Local Area Network Family 14-1
Professional Workstations 16-1
Smart FIX Product Repair Services 17-2
Specialized Product Center 17-10
System 320 Chassis for Multibus I 13-1
System 520 Chassis for Multibus II 4-1
System Enclosures & Backplanes 4-3
The Multibus II Bus Structure 5-1
UNIX OEM Developer Kit for Multibus II 1-23
Xpress Desktop Platforms 16-15
Xpress Disk Array Controller 16-32
Xpress/LX Deskside Platforms 16-26
Xpress/MX Server Platforms 16-21
XpressRack System 16-49

Multibus II Single Board Computers

1

iSBC 486/166SE AND iSBC 486/133SE SINGLE BOARD COMPUTERS



281140-1

HIGH-INTEGRATION CPU BOARDS: ONE BOARD DOES THE WORK OF THREE

The iSBC 486/166SE and iSBC 486/133SE accomplish in a single CPU board what previously required three Multibus II boards. Each board provides a 66 MHz Intel486™ DX2 or 33 MHz Intel486 microprocessor, SCSI and Ethernet interfaces, and up to 64 MB of fast-page DRAM.

FEATURES

- i486™ DX2 CPU at 66 MHz or i486 CPU operating at 33 MHz with on-chip FPU and cache
- Two 32 Mbyte modules may be stacked to offer 64 Mbyte of on-board DRAM
- 82258 ADMA operating at 10 MHz
- One Small Computer System Interface (SCSI) port
- SBX connector for expansion of on-board I/O
- Two 32-pin JEDEC DIP sites preconfigured with full MSA firmware in two 27020 EPROMs
- Two 32-pin PLCC sites that support EPROMs or Flash memory devices
- CSM/002 connector
- Parallel port—Centronics compatible
- 82530 serial communications controller offers two RS-232 ports
- On-board Ethernet using Intel 82596 LAN coprocessor at 33 MHz

iRMX® is a registered trademark and i386™, i387™, i486™, Intel386™, and Intel486™ are trademarks of Intel Corporation.
*Other brands and names are the property of their respective owners.

October 1993
Order Number: 281140-002

FEATURES

i486™ DX2 MICROPROCESSOR SPEED AND PERFORMANCE

The Intel486 DX2 Microprocessor and Intel486 DX Microprocessors are binary compatible with the Intel386™ Microprocessor. They provide the highest level of performance through state-of-the-art design using a pipelined architecture, 8 KB cache and a high-performance local bus interface. Frequent instructions execute in one cycle. The performance is further enhanced by an on-chip floating point unit (FPU) binary compatible with the Intel387™ Math Coprocessor.

WEALTH OF I/O

Through extensive use of surface mount and ASIC technology, the iSBC 486/166SE and iSBC 486/133SE boards increase the on-board I/O features compared to previous Multibus II CPU boards. The wealth of I/O features provides the user a variety of I/O interfaces in a single board. Slots are saved without limiting I/O.

- Ethernet — On-board using the Intel 82596 LAN Coprocessor running at 33 MHz
- SCSI — Single-ended SCSI interface through the P2 connector
- iSBX Bus Interface — Compliance Level: D16/16 DMA
- 2 Serial Ports — Based on Intel's 82530 Serial Communications Controller and RS-232-compatible
- Parallel Port — Centronics compatible

HIGH-SPEED MEMORY INTERFACE OPTIMIZES i486™ CPU PERFORMANCE

The iSBC 486/166SE and iSBC 486/133SE offer 8 MB to 64 MB of byte-parity protected fast-page DRAM which minimizes the memory access time, maximizing CPU performance.

HIGH-SPEED DMA

The 82258 Advanced DMA coprocessor provides 4 DMA channels. One channel is allocated to data transfers to/from SCSI and one channel is allocated to data transfers to/from the iSBX bus interface. The remaining two channels handle data transfers between the MPC and on-board memory. Special logic on the board allows the 82258 to transfer data to and from the MPC 4 bytes ("fly-by") or 16 bytes ("blast mode") at a time. Peak rates are 26 MB/sec in blast mode.

FLASH SUPPORT

Two 32-pin PLCC sites are available to support 512 KB of on-board programmable flash memory when using 28F020 devices. Flash may be used for user-specific firmware space or for non-volatile memory.

MULTIBUS II SYSTEM ARCHITECTURE (MSA)

MSA is a complete hierarchy of hardware, firmware and software interfaces and protocols that builds on and extends the basic Multibus II (IEEE/ANSI 1296) bus standard.

Specifically, MSA defines message passing, system boot and initialization, system and board-level diagnostics, and automatic board configuration in a multiprocessor system environment. By providing a standardized set of powerful programmatic interfaces and protocols, MSA assures a high level of vendor and board compatibility. This, in turn, provides OEMs faster time to market due to a shorter design/integration cycle.

The 32-pin JEDEC (DIP) EPROM sites are available to provide 1 MB of EPROM for 27040 4 MB devices. The sites, as shipped, contain BISTs (Built-In Self-Tests) and IDX (Initialization and Diagnostics eXecutive) power-up diagnostics and bootstrap loader residing in two pre-programmed 27020 EPROMS.

ON-BOARD CENTRAL SERVICES MODULE (CSM) CAPABILITY

An iSBC CSM/002 connector on the iSBC 486/166SE and iSBC 486/133SE provides an on-board CSM option. The iSBC CSM/002 module performs all CSM functions required by the IEEE/ANSI 1296 specification when the iSBC 486/166E or iSBC 486/133SE is installed in slot 0. The CSM also provides a battery-backed time-of-day clock, periodic alarm function and 28 bytes of non-volatile RAM.

intel

FEATURES

BUILT-IN MULTIBUS II COMPATIBILITY

The iSBC 486/166SE and iSBC 486/133SE are fully compatible with the iSBC 486/150, iSBC 486/125, iSBC 386/133 and iSBC 386/120 members of the Multibus II CPU family, as well as with the iSBC 386/258 SCSI Controller Board and iSBC 186/530 Ethernet Communication Engine. The iSBC 486/166SE and iSBC 486/133SE are capable of performing as a high performance CPU as well as a high performance SCSI or Ethernet controller.

The benefits that result from the high integration of the iSBC 486/166SE or iSBC 486/133SE include:

- Reduced cost by offering multiple system functions on one board
- Increased reliability through reduced board count
- Simplified sparing and service requirements

COMPREHENSIVE DEVELOPMENT AND OPERATING SYSTEM SUPPORT

Operating system support includes the 32-bit iRMX® III real-time operating system and UNIX* System V Release 4.0 operating system. The iRMK I real-time kernel is available for 32-bit embedded applications. All include Multibus II transport for full message-passing support.

INTEL QUALITY

All Intel boards are designed and manufactured to meet Intel's strict quality and reliability standards. Every process and procedure in our manufacturing flow is continuously monitored with statistical process control tools to ensure the highest level of quality.

ONE-YEAR WARRANTY

All Intel Multibus II boards have a one-year mail-in warranty on both parts and labor.

WORLDWIDE SERVICE AND SUPPORT

Should any Intel board ever need service beyond the warranty period, Intel has a worldwide network of service and repair facilities to keep you and your customers up and running.

1

SPECIFICATIONS

CPU CLOCK RATE

iSBC 486/166SE	i486 DX2 CPU @ 66.6 MHz
iSBC 486/133SE	i486 CPU @ 33.3 MHz

EPROM MEMORY

Model	Capacity
27010	256 KB
27020	512 KB
27040	1 MB

FLASH MEMORY

Model	Capacity
28F010	256 KB
28F020	512 KB

ENVIRONMENTAL CHARACTERISTICS

Operating Temperature:	0°C to 55°C (32°F to 131°F) at 200 LFM airflow across board (default configuration)
Relative Humidity:	0% to 90% (without condensation)
Storage Temperature:	−40°C to +70°C (−40°F to +158°F)

intel.

SPECIFICATIONS

INTERFACES

- Multibus II PSB
- iSBX Bus (compliance level D16/16 DMA)
- Serial I/O: RS-232-C DTE Async
- SCSI: single-ended
- Ethernet: 82596CA 32-bit LAN Coprocessor operating at 33 MHz
- Parallel port: Centronics compatible

PHYSICAL DIMENSIONS

iSBC 486/166SE and iSBC 486/133SE

Height:	233 mm (9.18 in)
Depth:	220 mm (8.65 in)
Front panel width:	19.2 mm (0.76 in)

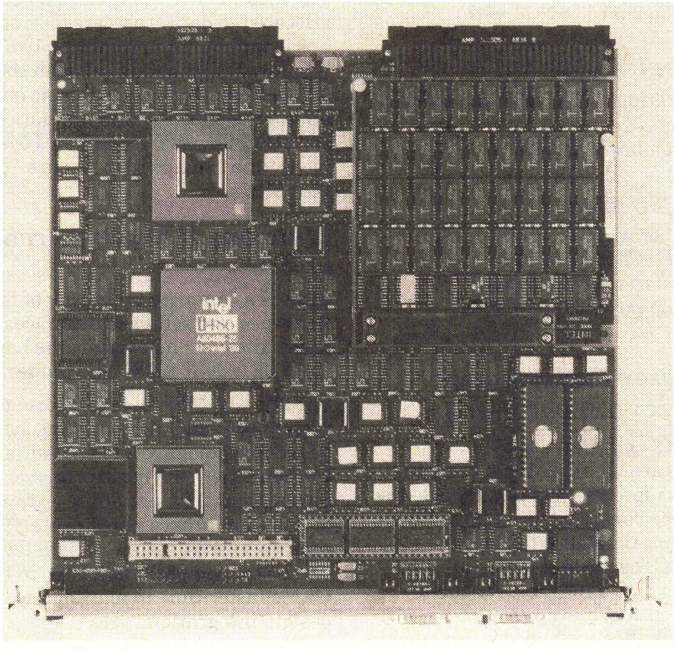
POWER REQUIREMENTS

iSBC486/166SE: +5V, 12.73A ±12V, 250 mA
iSBC486/1335SE: +5V, 12.5A ±12V, 250 mA

ORDERING INFORMATION

Order Code	Description
SBC486166SE008	66 MHz i486 DX2 CPU board with 8 MB DRAM, SCSI, and Ethernet
SBC486166SE016	66 MHz i486 DX2 CPU board with 16 MB DRAM, SCSI, and Ethernet
SBC486166SE032	66 MHz i486 DX2 CPU board with 32 MB DRAM, SCSI, and Ethernet
SBC486133SE008	33 MHz i486 CPU board with 8 MB DRAM, SCSI, and Ethernet
SBC486133SE016	33 MHz i486 CPU board with 16 MB DRAM, SCSI, and Ethernet
SBC486133SE032	33 MHz i486 CPU board with 32 MB DRAM, SCSI, and Ethernet
SBCMM3008	8 MB DRAM module
SBCMM3016	16 MB DRAM module
SBCMM3032	32 MB DRAM module

iSBC 486/150, iSBC 486/125 and iSBC 386/133 SINGLE BOARD COMPUTERS



281007-1

UPGRADE COMPATIBLE CPU BOARDS BOOST MULTIBUS II PERFORMANCE

The iSBC 486/150, iSBC 486/125 and iSBC 386/133 Single Board Computers provide immediate access to the Intel486™ DX2, Intel486 and Intel 386 32-bit Microprocessors on Multibus II, the industry standard multiprocessing system bus. Boasting the 50 MHz Intel 486 DX2, the 25 MHz i486 and 33 MHz i386 Processors and a feature set compatible with their predecessors, the iSBC 486/150, iSBC 486/125 and iSBC 386/133 Boards offer an immediate performance boost for current Multibus II designs.

FEATURES

iSBC 486/ 150, iSBC 486/125 and iSBC 386/133 CPU Boards

- 82258 ADMA with 16-byte "blast" mode
- iLBX II interface
- One iSBX connector
- Full 32-bit Multibus II (IEEE/ANSI 1296) Parallel System Bus interface
- Two 32-pin JEDEC DIP sites preconfigured with full MSA firmware in two 27020 EPROMs
- 82530 serial communications controller offers two RS-232 ports
- CMS/002 connector

iRMX® is a registered trademark and i386™, i387™, i486™, Intel386™, and Intel486™, are trademarks of Intel Corporation.
*Other brands and names are the property of their respective owners.

October 1993
Order Number: 281007-004

FEATURES

iSBC 486/150 CPU Board

- i486™ DX2 CPU operating at 50 MHz with on-chip FPU and cache
- 8 MB–64 MB on-board DRAM with parity

iSBC 486/125 CPU Board

- i486 CPU operating at 25 MHz with on-chip FPU and cache
- 8 MB–64 MB on-board DRAM with parity

iSBC 386/133 CPU Board

- i386™ CPU operating at 33 MHz
- i387™ Numeric Coprocessor
- 64 KB SRAM zero wait-state cache
- 1 MB–64 MB on-board DRAM with parity

Microprocessor Performance

i486™ and i486™ DX2 CPU Boards

The Intel486 DX2 and Intel486 Microprocessors are binary compatible with the Intel386 Microprocessor. They provide the highest level of performance through state-of-the-art design containing a pipelined architecture, 8 KB cache and high-performance local bus interface. Frequent instructions execute in one cycle. The performance is further enhanced by an on-chip floating-point unit (FPU) that is binary compatible with the i387 Numerical Coprocessor.

i386™ CPU Boards

The iSBC 386/133 Single Board Computer features an i386 Microprocessor running at 33 MHz. It also includes an i387 Floating Point Coprocessor running at 33 MHz. The 121 pin extended math coprocessor socket could also house a Weitek 3167 floating point unit (FPU) instead of the i387 FPU.

Balanced Set of On-Board I/O; Two serial Ports, iSBX Connector

Through extensive use of surface mount technology, the iSBC 486/150, iSBC 486/125, and the iSBC 386/133 Boards have increased on-board I/O features over previous Multibus II CPU boards. They provide two serial ports based on Intel's 82530 Serial Communications Controller and one iSBX connector, capable of supporting a single- or double-wide, 8- or 16-bit iSBX Multimodule board.

High-Speed DMA

The 82258 Advanced DMA coprocessor provides 4 DMA channels. Two channels are allocated to data transfers to/from the iSBX bus interface. The remaining two channels handle data transfers between the MPC and on-board memory Special logic on the board allows the 82258 to transfer data to and from the MPC 4-bytes ("fly-by") or 16-bytes ("blast" mode) at a time. Peak rates are 13.3 MB/s or 20.0 MB/s.

Multibus II System Architecture (MSA)

MSA is a complete hierarchy of hardware, firmware and software interfaces and protocols that builds on and extends the basic Multibus II (IEEE/ANSI 1296) bus standard.

Specifically, MSA defines message passing, system boot and initialization, system and board-level diagnostics, and automatic board configuration in a multiprocessor system environment. By providing a standardized set of powerful programmatic interfaces and protocols, MSA assures a high level of vendor and board compatibility This, in turn, provides OEMs faster time to market due to a shorter design/integration cycle.

The 32-pin JEDEC (DIP) EPROM sites are available to provide 1 MByte of EPROM for 27040 4-MBit devices. The sites, as shipped, contain BISTs (Built-In Self-Tests), IDX (Initialization and Diagnostics eXecutive), power-up diagnostics and bootstrap loader residing in two pre-programmed 27020 EPROMS.

On-Board Central Services Module (CSM) Capability

An iSBC CSM/002 connector on the iSBC 486/150, iSBC 486/125, and the iSBC 386/133 provide an on-board CSM option. The iSBC CSM/002 Module performs all CSM functions required by the IEEE/ANSI 1296 Specification. It also provides a battery backed time-of-day clock, periodic alarm function and 28 bytes of non- volatile RAM. The CPU board with CSM module is installed in slot zero of a Multibus II chassis and requires only one slot.

FEATURES

iLBX II Interface for Memory Mapped I/O Expansion

The iLBX interface on P2 provides expansion for 64 MBytes of off-board memory or memory mapped I/O. It operates at 8 MHz and is completely compatible with the iLBX II interface on the iSBC 286/100A.

Comprehensive Operating System Support

Operating system support includes the 32-bit iRMX® III real-time operating system and UNIX System V/386 operating system. The iRMK I real-time kernel is available for 32-bit embedded applications. All four include Multibus II transport for full message passing support.

High-Speed On-Board Memory

i486™ DX2 and i486™ CPU Boards

The iSBC 486/150 and iSBC 486/125 Boards come with 8 MB or 32 MB of byte-parity protected DRAM. On-board memory access is optimized via a two-way interleaved memory design. Moreover, the memory interface on the iSBC 486/150 and iSBC 486/125 Boards support up to 64 MB, allowing for further memory expansion.

i386™ CPU Boards

For optimum CPU performance, the iSBC 386/133 Board includes a 64 KB cache memory resulting in zero wait-state read accesses on cache hits. The iSBC 386/133 Board provides up to 64 MB of parity protected, fast page DRAM memory.

Full Compatibility

The iSBC 486/150 is fully compatible with its i386 and i486 Multibus II predecessors; namely, the iSBC 386/116, iSBC 386/120, iSBC 386/133, and iSBC 486/125. The iSBC 486/150 features, with the exception of PSB access to on-board memory, are a superset of the iSBC 386/116 and iSBC 386/120 feature set and matches that of the iSBC 386/133 and iSBC 486/125. Your i386 Multibus II designs can easily be upgraded to i486 performance with a simple board swap.

Worldwide Service, Support, and Training

Should these or any Intel board ever need service, Intel maintains a worldwide network of service and repair facilities to keep you and your customers up and running. For unique applications requiring customization of our products, the Intel Custom Board and Systems Group is available to modify, integrate, and test Intel boards and system components to your requirements.

Intel Quality—Your Guarantee

The iSBC 386/133, iSBC 486/150, and iSBC 486/125 Boards are designed and manufactured to meet Intel's strict quality standards, assuring their reliability and high quality.

intel

SPECIFICATIONS

CPU Clock Rate

iSBC 486/150:	i486 DX2 CPU @ 50 MHz
iSBC 486/125:	i486 CPU @ 25 MHz
iSBC 386/133:	i386 CPU @ 33.3 MHz i387 FPU @ 33.3 MHz

Cache Memory

iSBC 486/150:	8 KB on chip 0 wait state on read hit
iSBC 486/125:	8 KB on chip 0 wait state on read hit
iSBC 386/133:	64 KB extended 0 wait state on read hit

DMA Clock Rates/Maximum Bandwidth*

82258 ADMA 10 MHz/20 MB per second

*Note: Assumes transfer between local DRAM and MPC.

EPROM Memory

Two JEDEC sites provide the following capacity:

Devices	Capacity
27010	256 KB
27020	512 KB (supplied)
27040	1 MB

Interfaces

- Multibus II PSB: 32-bit Parallel System Bus (ANSI/IEEE 1296) interface with full message passing capability
- iSBX Bus: Compliance Level: D16/16 DMA
- iLBX Bus: Compliance Level: PROA
- Serial I/O: RS232C DTE ASYNC

Environmental Characteristics

Operating Temperature:	0°C to 55°C (32°F to 131°F) at 200 LFM airflow across board (default configuration)
Relative Humidity:	0% to 90% (without condensation)
Storage Temperature:	−40°C to +70°C (−40°F to +158°F)

Serial I/O Port

- Channel A & B:RS-232-C compatible DTE asynchronous interface
- 9-pin D-shell shielded connector
- Configurable baud rates: 300, 600, 1200, 2400, 4800, 9600, 19200, and 38400

Physical Dimensions

Height:	233 mm (9.18 inches)
Depth:	220 mm (8.65 inches)
Front Panel Width:	19.2 mm (0.76 inches)

Electrical Characteristics

DC Power Requirements (Typical)

iSBC486/150: +5V, 10.73A, ±12V, 200 mA

iSBC486/125: +5V, 10.50A, ±12V, 200 mA

iSBC386/133: +5V, 13.00A, ±12V, 200 mA

Note: Does not include power for iSBX module, or added iSBC MM0x modules.

For the second iSBC MM0x module, add:

iSBC MM01-FP or iSBC MM04-FP	+5V, 0.71A
iSBC MM02-FP or iSBC MM08-FP	+5V, 096A

ORDERING INFORMATION

Reference Manual

iSBC 386/133 Single Board Computer User's Guide (order number 457629)

iSBC 486/150 and iSBC 486/125 Single Board Computer User's Guide (order number 459660)

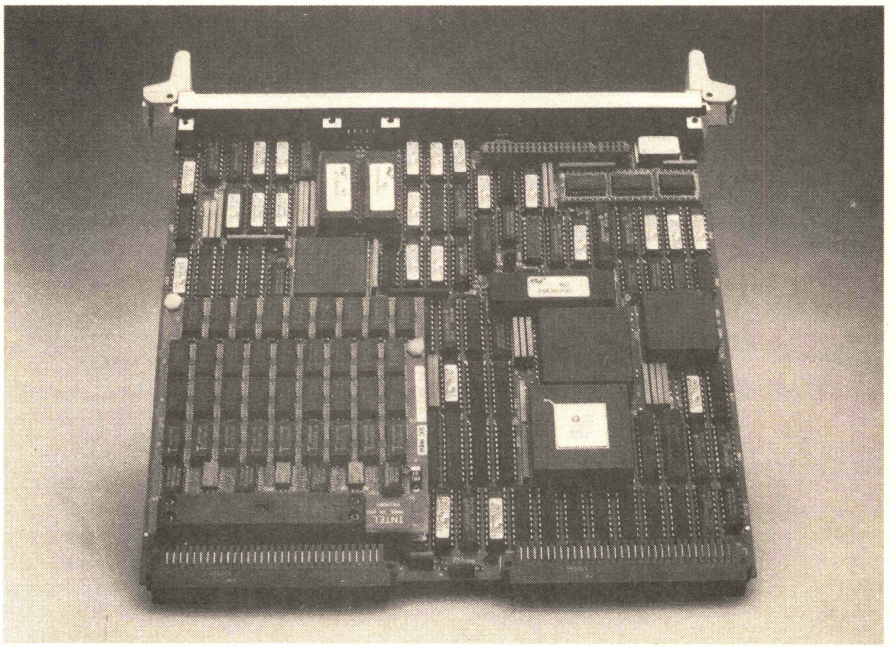
Order Code	Description
SBC486150C08	50 MHz i486 DX2 CPU Board with 8 MB DRAM
SBC486150C32	50 MHz i486 DX2 CPU Board with 32 MB DRAM
SBC486125C08	25 MHz i486 CPU Board with 8 MB DRAM
SBC486125C32	25 MHz i486 CPU Board with 32 MB DRAM
SBC386133F04	33 MHz i386 CPU Board with 4 MB DRAM
SBC386133F08	33 MHz i386 CPU Board with 8 MB DRAM
SBCMM08FC	8 MB memory expansion module for iSBC 486/125, iSBC 486/150
SBCMM32FC	32 MB memory expansion module for iSBC 486/125, iSBC 486/150
SBCMM01FP	1 MB memory expansion module for iSBC 386/133
SBCMM02FP	2 MB memory expansion module for iSBC 386/133
SBCMM04FP	4 MB memory expansion module for iSBC 386/133
SBCMM08FP	8 MB memory expansion module for iSBC 386/133
SBCMM16FP	16 MB memory expansion module for iSBC 386/133
SBCMM32FP	32 MB memory expansion module for iSBC 386/133
SBCCSM002	CSM Option Module

iSBC 386/120 MULTIBUS II SINGLE BOARD COMPUTER

- **High Performance 32-bit 386 Processor Operating at 20 MHz**
- **80387 Numerics Co-Processor Providing IEEE 754 Floating Point Instruction Set, Operating at 20 MHz**
- **64K byte Static RAM Cache Providing Zero Wait State Reads**
- **2, 4 or 8M Bytes of On-Board Dual-Ported Dynamic RAM Memory with Parity Error Detection, Expandable to 16M Bytes**
- **One RS 232C Serial I/O Port**
- **82258 DMA Controller Providing 4 High Performance DMA Channels**
- **32-Bit Multibus II Parallel System Bus (IEEE 1296) Interface with Full Message Passing Capability**
- **8-, 16-Bit iSBX Bus (IEEE P959) Interface with DMA for I/O Expansion**
- **Resident Firmware to Support Built-In-Self-Test (BIST) Power-Up Diagnostics**
- **Multibus II Interconnect Space for Software Configurability and Diagnostics**

The iSBC 386/120 Multibus II Single Board Computer is based on Intel's 386 high performance 32-bit microprocessor. The 386 CPU maintains software compatibility with the entire 8086 microprocessor family and delivers new performance standards for microcomputer-based systems. Three versions of the iSBC 386/120 board is offered: the M02, which contains 2M bytes of DRAM; the M04, which includes 4M bytes of DRAM; and the M08 which contains 8M bytes of DRAM. An optional memory expansion module can be added to expand the iSBC 386/120 board's resident memory to a maximum of 16M bytes.

The 64K byte static RAM cache enables the 386 CPU to execute at its full potential performance, while the Multibus II bus provides an interface for reliable, high performance multiprocessing.



280631-1

September 1993
Order Number: 280631-004

FUNCTIONAL DESCRIPTION

Overview

The iSBC 386/120 board utilizes Intel's 386 32-bit microprocessor. The advanced capabilities of the Multibus II architecture coupled with the high performance and compatibility features of the 386 CPU, provide the designer with a superior 32-bit solution for multiprocessing applications. By using the Multibus II architecture, multiprocessing systems are enhanced through advanced bus features including: 21-board distributed arbitration, virtual interrupts, hardware-assisted message passing, bus parity for high reliability, and software configurability using interconnect address space. The Multibus II parallel system bus (iPSB) interface on the iSBC 386/120 board supports full message passing and dual-port architectures and is fully compatible with other SBCs based on the Multibus II (IEEE 1296) bus specification.

The iSBC 386/120 board is offered in three versions: M02, M04 and M08 which contain 2, 4 and 8M bytes of resident DRAM memory respectively. This memory is physically located on an expansion board, and can be accessed directly from the iSBC 386/120 board's local bus or by another CPU over the iPSB bus. This dual-port memory can be expanded to a maximum of 16M bytes though the addition of a second Intel iSBC MM02FP, MM04FP or MM08FP (2, 4 or 8M byte) memory expansion module. Parity error detection is included on all resident DRAM memory.

Architecture

The iSBC 386/120 logic consists of eight resource modules and three interfaces connected together over an on-board local bus. The resources include the 386 CPU, the 80387 numeric co-processor, the 82258 DMA controller, the dual-port DRAM memory, the SRAM cache memory, the EPROM memory with BIST software, the programmable timers and the interrupt controllers. Interfaces included are the iPSB parallel system bus, the iSBX I/O bus and the RS 232C serial I/O interface. A block diagram of the iSBC 386/120 board is shown in Figure 1. The following text describes each of the resources and interfaces.

386 PROCESSOR

Intel's 386 CPU is the central processor for the iSBC 386/120 board. This is the first 32-bit member of Intel's 8086 family of microprocessors. At 20 MHz, the 386 is capable of executing at sustained rates of 5 million 32-bit instructions per second. This performance is made possible through a state-of-the-art design combining advanced VLSI semiconductor technology, a pipelined architecture, address translation caches and a high performance local bus interface.

The 386 processor provides a rich, generalized register and instruction set for manipulating 32-bit data and addresses. Features such as scaled indexing and a 64-bit barrel shifter ensure the efficient addressing and fast instruction processing. Special emphasis has been placed on providing optimized instructions for high-level languages and operating system functions. Advanced functions, such as hardware-supported multitasking and virtual memory support, provide the foundation necessary to build the most sophisticated multitasking and multiuser systems. Many operating system functions have been placed in hardware to enhance execution speed. The integrated memory management and protection mechanism translates virtual addresses to physical addresses and enforces the protection rules necessary for maintaining task integrity in a multiprocessing environment.

The 386 CPU provides access to the large base of software developed for the 8086 family of microprocessors. Binary code compatibility allows execution of existing 16-bit applications without recompilation or reassembly, directly in a virtual 8086 environment. Programs and even entire operating systems written for 8086 family processors can be run as tasks under 32-bit operating systems written for the 386 CPU.

80287 NUMERIC CO-PROCESSOR

The 80387 is a high-performance floating-point co-processor that takes numerics functions which would normally be performed in software by the 386 microprocessor and instead executes them in hardware. The instruction set executed by the 80387 is compatible with the IEEE 754 floating point standard, with high-precision 80-bit architectures and full support for single, double and extended precision operations. The 80387 executes floating point operations at a rate of 1.5M Whetstones per second at 16 MHz, and 1.86M Whetstones per second at 20 MHz.

82258 ADVANCED DMA CO-PROCESSOR

The 82258 is a high performance 4 channel DMA co-processor. Unlike other DMA devices, the 82258 has processing capabilities. Its command chaining feature and data manipulation capabilities (compare, verify, translate), allow the 82258 to execute simple

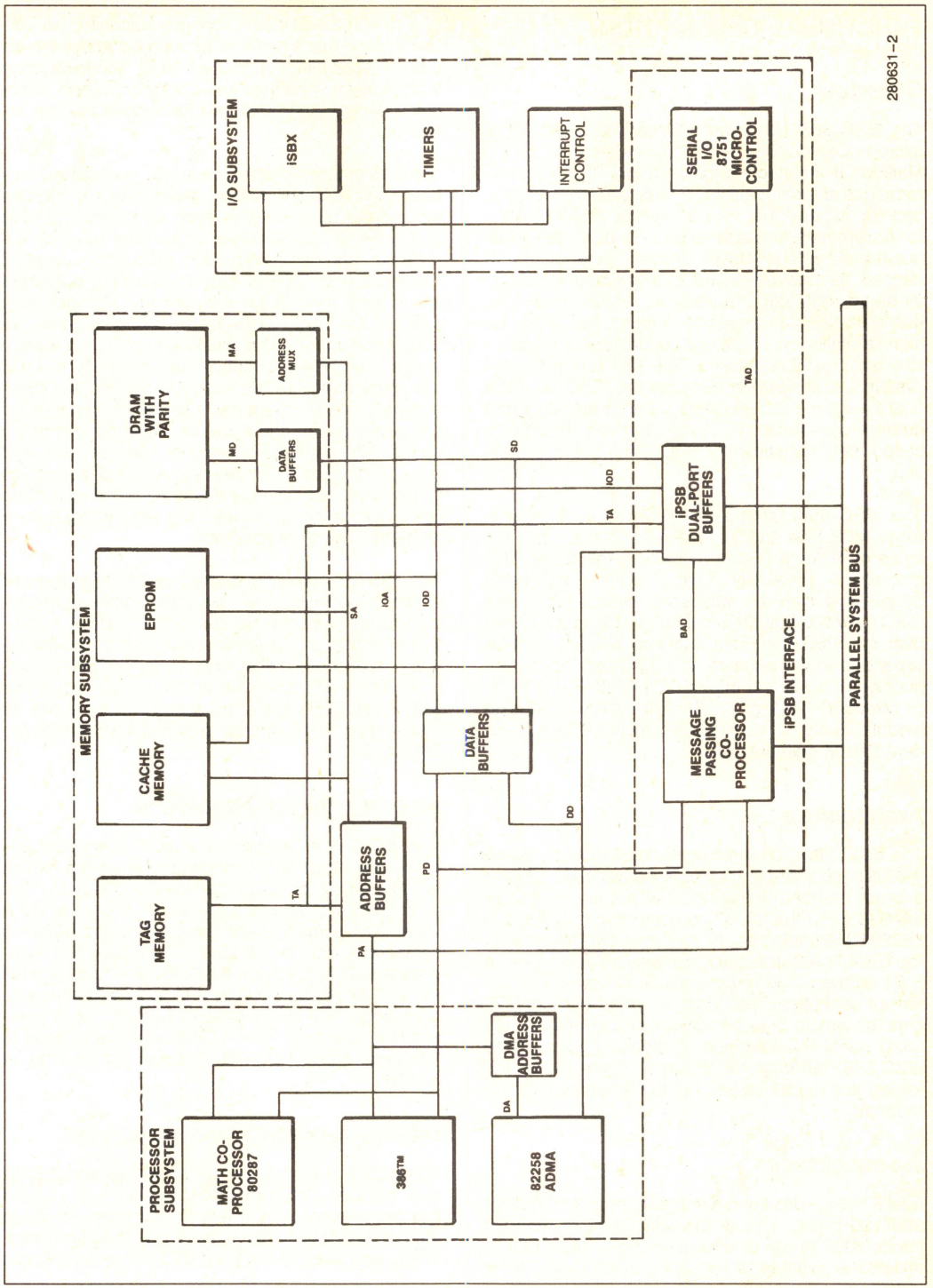


Figure 1. Block Diagram of the iSBC 386/120 Board

input/output programs without processor intervention. This allows the 386 CPU more time for general purpose processing, thus improving total system performance. The 82258 shares ownership of the on-board local bus via the 386 processor's HOLD, HOLD ACKNOWLEDGE co-processor protocol. The maximum percentage of on-board local bus utilization by the 82258 is user programmable.

The four 82258 channels are allocated to iSBC 386/120 on-board resources as shown in Table 1. Special logic on the boards allows the 82258 to transfer data to and from the message passing coprocessor (MPC) 32-bits at a time using single cycle mode. Using this mode, the 82258 (which operates at 10 MHz on 386/120) can load or unload an MPC solicited message (from or to resident DRAM) at a sustained rate of 10.7M bytes and 13.3 Mbytes per second, respectively.

Table 1. DMA Channel Allocation

Channel	Function
0	iSBX DMA support
1	iSBX DMA support
2	MPC Solicited Message Receive
3	MPC Solicited Message Transmit

DUAL-PORTED DYNAMIC RAM

The iSBC 386/120 board includes 2, 4 or 8M bytes of DRAM depending upon the version. This memory can be extended to a maximum of 16M bytes through the addition of an Intel iSBC MM02, MM04 or MM08: 2, 4 or 8M byte memory expansion module. The DRAM refresh control, dual-port control and parity generation/checking logic is physically located on the baseboard, while the actual DRAM components are located on low-profile surface mount expansion boards. Each iSBC 386/120 board is shipped with one expansion memory module installed and may be expanded to contain two total memory expansion modules. The memory expansion module mechanics are shown in Figure 2.*

***NOTE:**
Only one single-sided memory module (MM04FP) installed onto the iSBC 386/120 board will fit within one Multibus II slot. A double-sided module (MM02FP or MM08FP) or any stack of two modules will require two Multibus II slots.

Parity error detection is provided on a byte-by-byte basis. The parity logic normally generates and checks for odd parity with detected errors signaled via an on-board LED and a CPU interrupt. Even parity can be forced to generate a parity error for diagnostic purposes.

The DRAM is accessible from both the on-board local bus and the iPSB bus. The amount of memory accessible from the iPSB bus and the iPSB address aliasing values are dynamically configurable via interconnect space registers.

CACHE MEMORY

The cache memory on the iSBC 386/120 board allows zero wait-state accesses to memory when the data requested is resident in the cache memory. The static RAM cache has 16,384 32-bit data entries with 8-bit "tag" fields. Each 32-bit DRAM memory location maps to one (and only one) cache data entry. The "tag" fields are used to determine which 32-bits of DRAM memory currently resides in each cache data entry. The combination of a direct mapped cache data array and a tag field ensures data integrity and accurate, high performance identification of cache "hits".

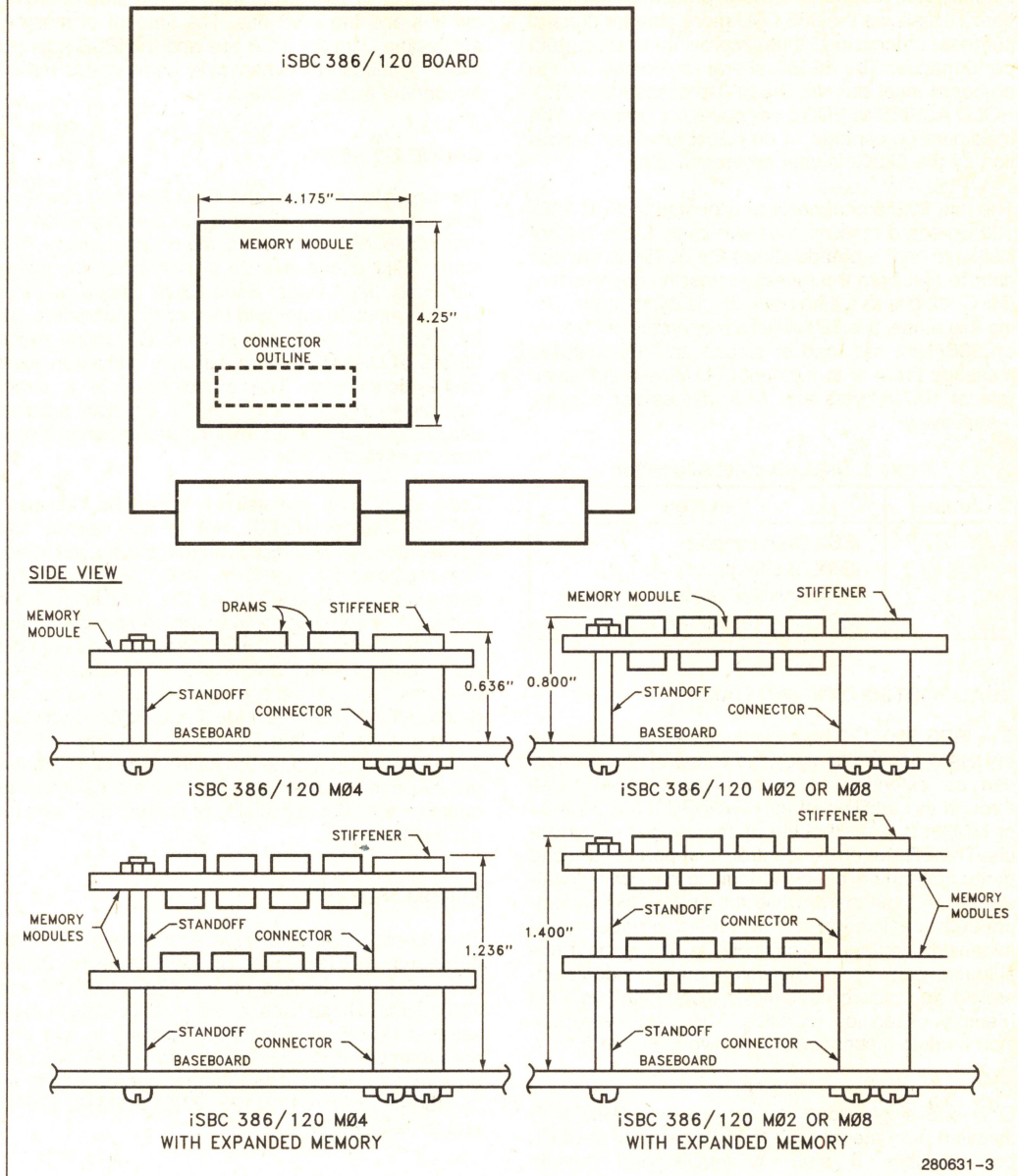
Data integrity is maintained for cache "misses" (DRAM memory READs not in the cache) and DRAM memory WRITEs through a simple, yet effective replacement algorithm. 386 CPU generated cache READ "misses" cause the data field of the cache entry corresponding to the addressed memory to be filled from the DRAM array and the tag field to be updated. All iPSB or ADMA READs are treated as cache "misses", except that the cache is not updated. All WRITE "hits", local and iPSB generated, cause the cache data field to be updated. WRITE "misses" do not update the cache. The cache memory size and replacement algorithm are designed to optimize both the probability of cache "hits" and local bus utilization.

EPROM MEMORY

Two 32-pin JEDEC EPROM sites capable of supporting up to 512K bytes of EPROM (using 27020 EPROMs) are supplied on the iSBC 386/116 and 120 boards. These sites, as shipped, contain built-in-self-test power-up diagnostics residing in two preprogrammed 27512 EPROMs. These EPROMs may be replaced by the user. Jumper configurations allow the use of 2764, 27128, 27256, 27512, 27010, and 27020 EPROMs.

8254 PROGRAMMABLE TIMERS

The iSBC 386/120 board contains an Intel 8254 component which provides three independent programmable 16-bit interval timers. These may be used for real-time interrupts or time keeping operations. Outputs from these timers are routed to one of the two 8259A interrupt controllers to provide software programmable real-time interrupts.



NOTE:
1. The iSBC 386/120 M04 fits within one MULTIBUS II slot. The iSBC 386/120 M02 or M08 with a stack of two memory modules will require two MULTIBUS II slots.

Figure 2. iSBC 386/120 Board Memory Module Mechanics

INTERRUPT CONTROL

Two Intel 8259A programmable interrupt controllers on the iSBC 386/120 board is used in a master-slave configuration for prioritizing up to 15 separate on-board interrupt sources. The devices and functions are listed in Table 2.

The Multibus II iPSB bus utilizes virtual interrupts (called unsolicited messages) for board-to-board signaling. The bus interface component (MPC) queues-up incoming virtual interrupts from the iPSB bus and generates a single message interrupt (MINT) signal. This signal is connected into one of the 8259A interrupt controllers for prioritization and interruption of the host 386 CPU. Error conditions occurring on the iPSB bus will cause the MPC to generate an error interrupt (EINT) signal. This signal is connected to another 8259A interrupt input.

Other interrupt sources come from the 82258 DMA controller, the 8254 timers, the iSBX interface, the 8751 serial port, and the DRAM parity checker.

SERIAL I/O INTERFACE

One RS 232C compatible serial I/O port is provided via the Intel 8751 microcontroller. This port is configured as a data terminal equipment (DTE) asynchronous serial port. Mechanically, the serial port exits through the iSBC 386/120 board's front panel via a 9-pin D-shell connector.

iPSB BUS INTERFACE

The Multibus II parallel system bus interface is implemented by Intel's MPC (message passing coprocessor) and a pre-programmed 8751 microcontroller. This interface supports full arbitration, transfer and error checking features as defined in the iPSB specifications. In addition, the interface supports advanced features of the iPSB bus including hardware message passing and autoconfiguration through geographic addressing.

The MPC component contains nine 32-byte buffers which are used to decouple iPSB bus traffic from iSBC 386/120 local bus traffic through the concept known as message passing. These nine buffers are utilized as follows: four buffers queue-up incoming unsolicited messages, one buffer stores an out-going unsolicited message, two buffers are used to double-buffer an out-going solicited message, and two buffers are used to double-buffer an incoming solicited message. These buffers are capable of transferring data packets over the iPSB bus at its maximum transfer rate. Unsolicited messages include address and type fields and 28 bytes of user-defined data, and are transferred over the iPSB bus in 900 ns. Solicited messages are automatically divided into small packets, with each packet containing address and type fields and 32 bytes of user-defined data. Each solicited message packet is transferred over the iPSB bus in 1000 ns.

Table 2. 8259A Interrupt Sources

Device	Function	Number of Interrupts
MPC-MINT	Signals arrival of virtual interrupt over iPSB bus, solicited input complete, transmit FIFO not full or transmit error	1
MPC-EINT	Signals error condition on the iPSB bus	1
82258 DMA	Transfer complete	1
8254 Timers	Timers 0, 1, 2 outputs, function determined by timer mode	3
8751 Serial Port	Serial diagnostic port requests	1
iSBX Interface	Function determined by iSBX bus multimodule board	4
DRAM Parity Checker	Signals parity error	1

The 8751 component implements the iPSB geographic addressing feature called Interconnect space. Read-only registers are used to hold information such as board type and revision level. Software configurable registers are used for auto-configurability, local or remote diagnostics and software controlled reset. In addition, the 386 CPU executes power-up built-in self tests of the various resources on the iSBC 386/120 board. The results of these tests are reported via registers in interconnect space. After successfully completing its BIST routines, the 386 CPU must clear the reset-not-complete register. If, after 30 seconds, the reset-not-complete has not been cleared, the 8751 resets the local bus and holds it in a reset state. In this way, only a few components on the iSBC 386/120 board must be functional to allow the iPSB bus to operate.

iSBX BUS INTERFACE

One iSBX connector, capable of supporting one single- or double-wide, 8- or 16-bit iSBX MULTIMODULE board, is provided on the iSBC 386/120 board for the addition of an optional I/O module. Two DMA channels from the 82258 can be used with iSBX modules which require DMA support.

SPECIFICATIONS

Word Size

Instruction — 8-, 16-, 24-, 32-, 40-bit
Data — 8-, 16-, 32-bit
Floating Point Data — 80-bit

Clock Rates

	386/120
386 CPU	20 MHz
80387 Numeric Co-processor	20 MHz
82258 DMA	10 MHz

Dual-Port DRAM Memory

DEFAULT CAPACITY

iSBC 386/120 M02—2M byte
iSBC 386/120 M04—4M byte
iSBC 386/120 M08—8M byte

EXPANSION MODULES

iSBC MM02FP—2M byte
iSBC MM04FP—4M byte
iSBC MM08FP—8M byte

MAXIMUM CAPACITY—16M BYTES

EPROM Memory

Default — 128K byte using two pre-programmed 27512 EPROMs
Capacity — Two 24-, 28- or 32-pin JEDEC-compatible devices

EPROM	Memory Capacity
2764	16 KB
27128	32 KB
27256	64 KB
27512	128 KB
27010	256 KB
27020	512 KB

Timers

Capability — Three independently programmed 16-bit interval timers
Input Frequency — 1.25 MHz ±0.1%
Output Period — 1.6 μs to 52.4 ms

Interrupt Capability

Incoming Interrupts — 255 individual and 1 broadcast from iPSB bus 12 local sources (see Table 2)
Outgoing Interrupts — 255 individual and 1 broadcast to IPSB bus

Serial Port Interface

RS 232C Electrical
Asynchronous, DTE only
9-pin D-shell connector
Baud rates: 9600, 4800, 2400, 1200, 300, 110 bits/sec

iSBX Interface

Capability — One 8- or 16-bit, single- or double-wide iSBX module
Compliance Code — D16/16 DMA

iPSB Interface

Capability— Requesting and replying agent supporting 8-, 16-, 24- and 32-bit transfers, parity bit generation and checking, unsolicited and solicited message passing, and autoconfiguration through interconnect space.

Physical Dimensions

Length:	220 mm (8.6 in.)
Width:	233 mm (9.2 in.)
Front Panel Height:	19.2 mm (0.76 in.)

Power Requirements

5V: 11.14 Amps
12V: 0.046 Amps
−12V: 0.041 Amps
Voltage tolerance ±5%

Temperature Range and Airflow Requirements

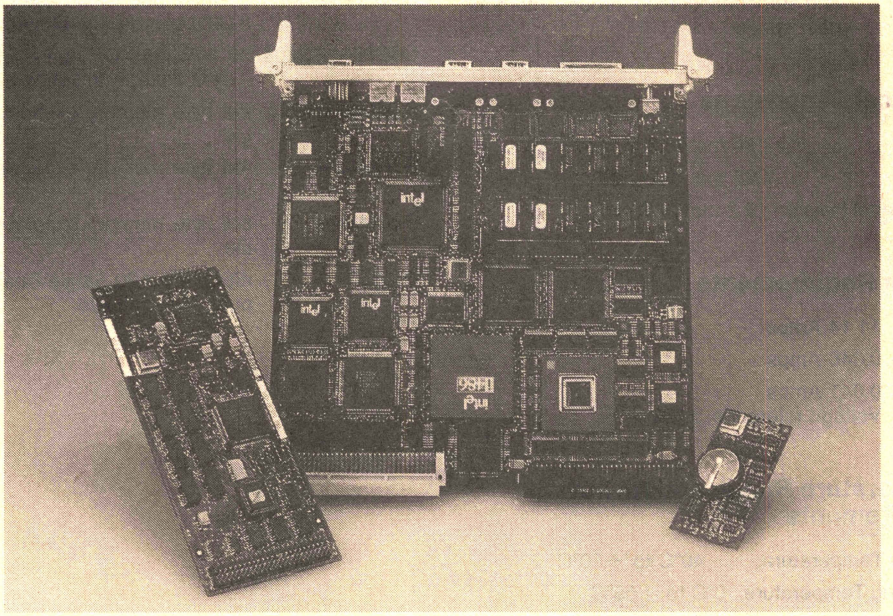
Storage Temperature:	−40°C to +70°C
Operating Temperature:	0°C to +55°C
Airflow:	200 LFM minimum

ORDERING INFORMATION

Part Number	Description
SBC386120M02	20 MHz 386 CPU-based Multibus II CPU board with 2M byte memory
SBC386120M04	20 MHz 386 CPU-based Multibus II CPU board with 4M byte memory
SBC386120M08	20 MHz 386 CPU-based Multibus II CPU board with 8M byte memory
SBCMM02FP	2M byte memory expansion module
SBCMM04FP	4M byte memory expansion module
SBCMM08FP	8M byte memory expansion module
451833	iSBC 386/120 Single Board Computer Users Guide

1

iSBC 486/DX66, iSBC 486/DX33, AND iSBC 486/SX25 PC-COMPATIBLE BOARD



281374-1

A POWER PC ON A POWER BUS

The iSBC 486/DX66, iSBC 486/DX33, and iSBC 486/SX25 boards bring the power and integration of today's high-end personal computers to Multibus II, the industry-standard, cooperative multiprocessing system bus. The marriage of high-end personal computer graphics, integrated I/O, and standard software combined with the multiprocessing, high-bandwidth power of the Multibus II Architecture allows you quick time to market with a system based on industry standards. The iSBC 486/DX66, iSBC 486/DX33, and iSBC 486/SX25 can be used as the front-end of a Multibus II system or as a compute engine.

FEATURES

- Completely integrated PC with graphics, I/O, and memory in a single Multibus II slot
- 100% PC/AT compatible
- Intel486™ DX2 CPU operating at 66 MHz or an Intel486 CPU at 33 MHz, with on-chip Floating Point Unit (FPU) and cache, or an Intel486 SX CPU operating at 25 MHz
- Up to 32 MBytes of on-board SIMM DRAM with parity protection
- IDE (Intelligent Drive Electronics) or SCSI (Small Computer Systems Interface) peripheral interfaces
- Super VGA graphics via an optional add-on module

iRMX® is a registered trademark and Intel386™, Intel387™, and Intel486™ are trademarks of Intel Corporation.
*All other names and marks are the property of their respective owners.

October 1993
Order Number: 281374-003

FEATURES

- Standard PC Ethernet networking via an optional add-on module
- Industry standard AT BIOS and Multibus Systems Architecture firmware in Intel Flash memory
- P2/aPC (ISA) bus on P2 connector
- Supports DOS, iRMX® for Windows,* and UNIX* System V Release 4.0 Version 3
- Front-panel recessed reset switch
- iSBC CSM/002 support

Intel486™ DX2 MICROPROCESSOR SPEED AND PERFORMANCE

The Intel486 DX2 and Intel486 microprocessors are binary compatible with the Intel386™ microprocessor. They provide the highest level of performance through state-of-the-art design using a pipelined architecture, 8 KB cache. and a high-performance local-bus interface. Frequent instructions execute in one cycle. The performance is further enhanced by an on-chip floating point unit (FPU) that is binary compatible with the Intel387™ numeric coprocessor.

100% PC/AT-COMPATIBILITY AND A FULL COMPLEMENT OF I/O

The iSBC 486/DX66, iSBC 486/DX33, and iSBC 486/SX25 are 100% PC/AT compatible. The boards undergo the same rigorous compatibility testing that commercial PC products endure—this means you can be assured that 100% PC/AT compatibility is achieved.

The iSBC 486/DX66, iSBC 486/DX33, and iSBC 486/SX25 support a wealth of I/O. The Multibus II P2 connector brings out the P2/aPC (ISA) bus, the IDE or SCSI peripheral interface, and the floppy disk interface. The front panel provides two serial ports, one parallel port, a SuperVGA port or PC Ethernet port, a keyboard port, and a mouse port.

OPERATING SYSTEM SUPPORT

Because the iSBC 486/DX66, iSBC 486/DX33, and iSBC 486/SX25 are 100% PC/AT compatible, operating systems that run on the PC/AT architecture will run on this board. In addition, DOS, iRMX for Windows, and UNIX System V Release 4 have enhanced Multibus II support via OEM developer's kits (MBIIDOSODK and MBIIUNIXODK). External connectivity is provided using the standard protocols like TCP/IP, UDP, INA, and IPX/SPX. Based on these protocols network products like RFS, NFS, and OpenNET are supported on UNIX, and Novell* is supported on DOS. TLI and TAI interface support is also provided on UNIX. The MBIIDOSODK provides board to board communication via NetBIOS and DDE commands for boards running DOS, Windows, or iRMX for Windows. MBIIUNIXODK provides board to board communication via Multibus II transport for boards running UNIX.

P2 BACKPLANES AND AT/XT CARRIER CARD*

Not only do the iSBC 486/DX66, iSBC 486/DX33, and iSBC 486/SX25 allow you to use standard software, they also accommodate the expansion cards. The P2 backplanes serve three purposes: they provide the IDE or SCSI cable connection, the floppy drive cable connection, and give you the ability to add AT or XT expansion cards. The P2 backplanes come in either 1, 2, or 4-slot versions.

The AT/XT Carrier Card requires two Multibus II slots and allows you to add one standard half-length AT or XT expansion card. For example, you could add an Ethernet card to connect to a network or an ISA bus extension card to connect with another ISA chassis for additional ISA card expansion.

MULTIBUS II SYSTEMS ARCHITECTURE (MSA) FIRMWARE

MSA is an open standard defining a complete hierarchy of hardware, firmware, and software interfaces and protocols that build on and extend the basic Multibus II (IEEE/ANSI 1296) bus standard. Specifically, MSA defines message passing, system boot and initialization, system and board level diagnostics, and automatic board configuration in a multiprocessor system environment. By providing a standardized set of powerful programmatic interfaces and protocols, MSA assures a high level of inter-vendor compatibility at the system level. This, in turn, provides OEMs faster time to market due to a shorter design/integration cycle.

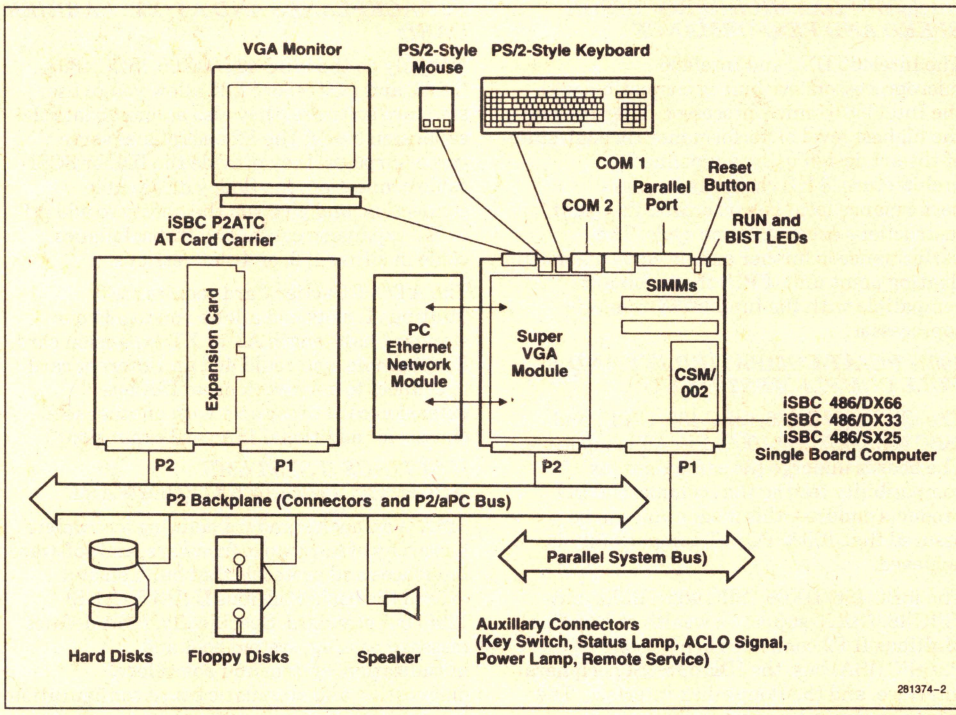
ON-BOARD CENTRAL SERVICES MODULE (CSM) CAPABILITY

An iSBC CSM/002 connector on the iSBC 486/DX66, iSBC 486/DX33, and iSBC 486/SX25 provides an on-board CSM option. The iSBC CSM/002 module performs all CSM functions required by the IEEE/ANSI 1296 specification when the iSBC 486/DX66, iSBC 486/DX33, or iSBC 486/SX25 is installed in slot 0. These functions include central Multibus II clock

generation, continuous bus monitoring, and error detections. The CSM also provides system-wide, battery-backed time-of-day clock, periodic alarm function, and 28 bytes of non-volatile RAM.

ONE-YEAR WARRANTY AND WORLDWIDE SERVICE AND SUPPORT

All Intel board-level products have a one-year mail-in warranty on both parts and labor via our worldwide service and support network. Should any Intel board ever need service beyond the warranty period. Intel maintains a worldwide network of service and repair facilities to keep you and your customers up and running. In addition, our international Systems Engineering organization is available to integrate Intel boards and system components into your products.



iSBC 486/DX66, iSBC 486/DX33, and iSBC 486/SX25 PC Subsystem Block Diagram

SPECIFICATIONS

BOARD

CPU	Intel486 DX2 CPU @66 MHz, Intel486 DX CPU @33 MHz, or Intel486 SX CPU @25 MHz
ON-BOARD MEMORY	2 SIMM sockets, 25.5mm height
Maximum RAM	32 MBytes
Upgrade RAM	256K x 26, 512K x 36, 1M x 36, 2M x 36, and 4M x 36
Access Time	80 ns or faster
Error Detection	Parity

INTEGRATED GRAPHICS

Controller	Western Digital WD90C30
Video Memory	1 MB (70 ns DRAM)
Maximum Resolution	1024 x 768, 256 colors
Compatible with	SVGA, VGA, EGA, CGA, HGC, MDA
Video Port	15-pin connector; analog

I/O

2 Serial Ports	Async, RS-232, 9-pin
1 Parallel Port	Centronics-compatible, 25-pin
1 Mouse Port	PS/2 compatible
1 Keyboard Port	PS/2 compatible
Floppy Disk Controller	Intel 82077AA-1
Peripheral Interface	IDE or SCSI
P2 Connector	ISA Bus (P2/aPC specification)

PHYSICAL CHARACTERISTICS

STANDARD MULTIBUS II BOARD

The iSBC 486/DX66, iSBC 486/DX33, and iSBC 486/SX25 require only a single slot in a Multibus II Chassis, even with 32 MBytes of DRAM, the SuperVGA module or PC Ethernet network module, and the iSBC CSM/002 installed.

PHYSICAL DIMENSIONS

Height:	233mm (9.18 in)
Depth:	220mm (8.65 in)
Front Panel Width:	19.2mm (0.76 in)

ELECTRICAL CHARACTERISTICS AND ENVIRONMENTAL REQUIREMENTS

POWER REQUIREMENTS

DC Power Requirements (typical):	4.5A at 5 V_{DC} 100 mA + 12 VDC

ENVIRONMENTAL

Storage Temperature:	−40°C to +70°C (−40°F to +158°F)
Operating Temperature:	0°C to +55°C (32°F to 131°F)
Storage Humidity:	5%–95% non-condensing at 55°C (131°F)
Operating Humidity:	8%–90% non-condensing at 55°C (131°F)

REFERENCE MANUALS

The following manuals provide necessary information when using the ISBC 486/DX66, iSBC 486/DX33, or iSBC 486/SX25 board:

iSBC 486/DX66, iSBC 486/DX33, and iSBC 486/SX25 Single Board Computer User's Guide, Order Number: 469026

Multibus II Interconnect Interface Specification, Order Number: 149299

Introduction to Multibus II System Architecture, Order Number: 508305

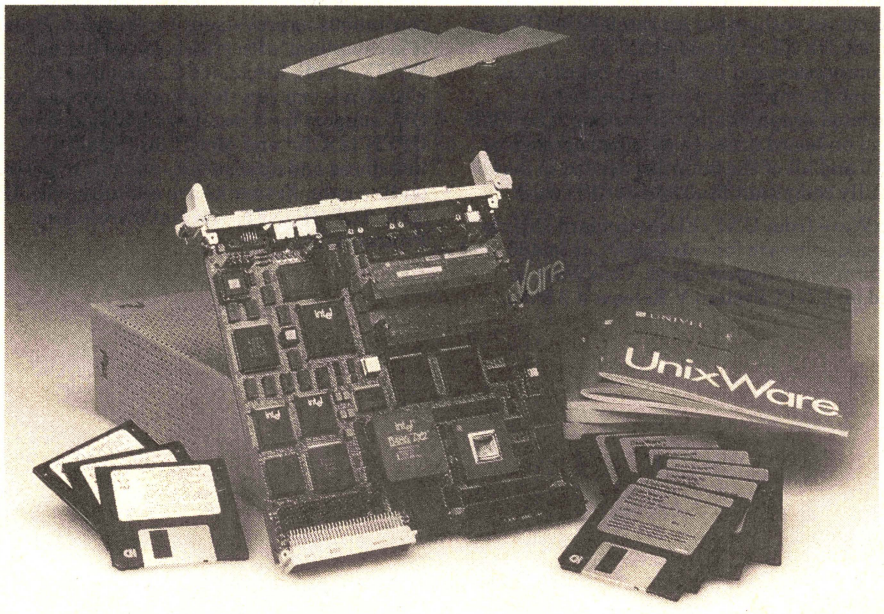
Firmware User's Guide for Multibus II System Architecture (MSA) Firmware, Order Number: 506090

Intel P2/aPC Bus Specification, Order Number: 500797

ORDERING INFORMATION

Order Code	Description
SBC486DX6600V	66 MHz Intel486 DX2, 0 MB DRAM, SuperVGA graphics, one-slot IDE P2 backplane
SBC486DX6600	66 MHz Intel486 DX2, 0 MB DRAM, one-slot IDE P2 backplane
SBC486DX3300V	33 MHz Intel486 CPU, 0 MB DRAM, SuperVGA graphics, one-slot IDE P2 backplane
SBC486DX3300	33 MHz Intel486 CPU, 0 MB DRAM, one-slot IDE P2 backplane
SBC486SX2500V	25 MHz Intel486 SX CPU, 0 MB DRAM, SuperVGA graphics, one-slot IDE P2 backplane
SBC486SX2500	25 MHz Intel486 SX CPU, 0 MB DRAM, one-slot IDE P2 backplane
SITP2APCIDE2	Two-slot IDE backplane for connecting to one SBCP2ATC board
SITP2APCIDE4	Four-slot IDE backplane for connecting to two SBCP2ATC boards
SITP2APCSCSI1	One-slot SCSI backplane
SITP2APCSCSI2	Two-slot SCSI backplane for connecting to one SBCP2ATC board
SITP2APCSCSI4	Four-slot SCSI backplane for connecting to two SBCP2ATC boards
SBCP2ATC	Two-slot XT or AT carrier card
MBIIDOSODK	OEM developer's kit for customers running DOS/Windows or iRMX for Windows
MBIIUNIXODK	OEM developer's kit for customers running UNIVEL's UnixWare UNIX System V Release 4.2
SBCCSM002	Central Services module
SBCEWENET	Standard PC Ethernet networking module. Mounts in place of VGA module.

UNIX* OEM DEVELOPER KIT FOR MULTIBUS II



281147-1

SHRINK-WRAPPED PC UNIX AND STANDARD PC HARDWARE ON A POWERFUL BUS

MBIIUNIXODK is the glue that binds Intel's Multibus II Embedded WorkStation and standard UnixWare* from UNIVEL* into a complete and robust solution which provides flexibility in design, complete integration and system communication. The resulting system is powerful, easily expandable and based on open standards.

MBIIUNIXODK adds support for Multibus II backplane communication to standard shrink-wrapped UNIVEL UnixWare. MBIIUNIXODK provides three add-on packages; Multibus II transport, Multibus II board drivers, and transport applications interface (TAI). Multibus II transport includes the transport kernel interface (TKI) and interconnect support for high-speed communication between CPU boards. The board drivers include support for Ethernet, I/O, and peripheral boards. The TAI allows the Embedded WorkStation to download applications to other Multibus II CPU boards. In summary, the MBIIUNIXODK allows an Embedded WorkStation running UNIVEL's UnixWare to communicate with other CPUs on the Multibus II backplane running iRMX®, iRMK, DOS, Windows or any OS with Multibus II transport capability.

iRMX® is a registered trademark and Intel386, Intel486 and Pentium are trademarks of Intel Corporation.
*All other names and marks are the property of their respective owners.

October 1993
Order Number: 281147-001

intel

FEATURES

The Embedded WorkStation (EWS) is a fully PC-compatible, Multibus II CPU board. The EWS comes with either an Intel486™ DX2 66, Intel486 DX 33, or Intel486 SX 25 microprocessor and includes on-board SCSI, IDE, and floppy controllers as well as integrated graphics and networking. The EWS has an on-board message passing coprocessor (MPC) and all of the facilities needed to make it a fully compliant Multibus II CPU card.

UnixWare from UNIVEL is standard shrink-wrapped software for Intel386™, Intel486, and Pentium™ microprocessors. UnixWare is based on UNIX System V Release 4.2 from USL, a powerful and highly integrated 32 bit operating system which has undergone continuous improvement for 20 years. Because it is compliant with standards including POSIX, FIPS, XPG3, iBCS2, and iABI, UnixWare can run thousands of applications. GUI support for X-windows is provided by OPEN LOOK* and Motif* for friendly interfaces and applications such as hypertext on-line help. Networks are seamless with IPX/SPX, TCP/IP, NFS, and NetWare* Core Protocols (NCP).

Multibus II Memory Expansion Boards

2

HIGH PERFORMANCE MEMORY MODULES

- **Provides Fast Page SMT Memory Expansion for Intel's iSBC 386/12, iSBC 386/133, iSBC 386/258, Mix Baseboards iSBC 486/12 and SBC 486/133SE Boards**
  - **— Also Available SMT Memory Expansion Modules for the iSBC 486/125 Boards**
- **Available in 1M, 2M, 4M, 8M, 16M and 32M Byte Sizes**
- **32 Bits Wide with Byte Parity**
- **Stackable to Provide up to 64 MBytes of High Speed Memory for MULTIBUS I and MULTIBUS II Boards**
- **Supports 32-Bit, 16-Bit and 8-Bit Data Paths**
- **Supports Independent Read/Writes**
- **Easily Installed**

The iSBC MM0XFP DRAM memory modules are members of Intel's complete line of iSBC memory and I/O expansion boards. The MM-Series of memory modules use a dedicated memory interface to maximize CPU/memory performance.

2

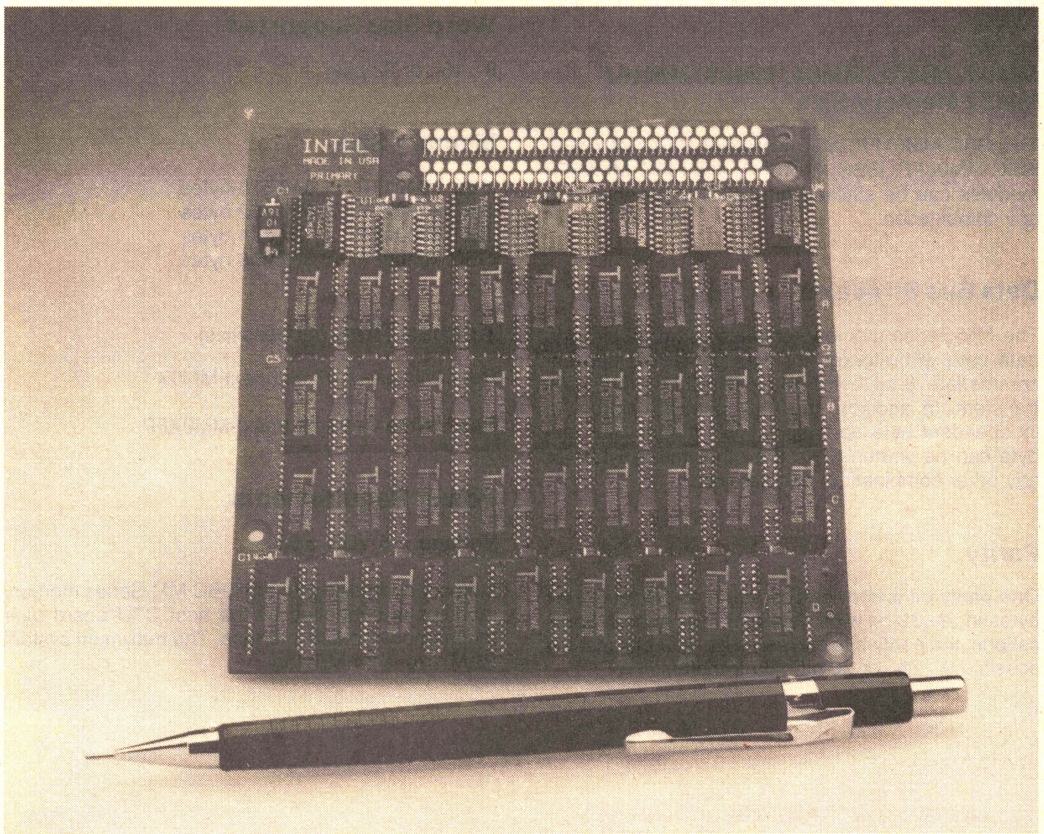


Figure 1. iSBC MM08FP Memory Module

281010-1

*The iSBC MM01FP, MM02FP, MM04FP, MM08FP, MM16FP, MM32FP memory modules are also manufactured under product code pSBC MM01FP, MM02FP, MM04FP, MM08FP, MM16FP, MM32FP by Intel Puerto Rico, Inc.

October 1992
Order Number: 281010-002

FUNCTIONAL DESCRIPTION

The iSBC MM-Series provide high performance, 32-bit parity DRAM memory for the MULTIBUS I and MULTIBUS II boards. These CPU boards come standard with one MM-Series module installed, with memory expansion available through the addition of a second stackable iSBC MM-Series module.

Memory Access Capabilities

The dynamic RAM memory of the memory modules is accessed through the dedicated memory module interface.

The MM memory module is designed for direct transfer of data between the CPU and the memory module without accessing the MULTIBUS interface.

MM01/MM02/MM04/MM08/MM16/ MM32 Memory Size

The iSBC MM01FP, iSBC MM02FP, iSBC MM04FP, iSBC MM08FP, iSBC MM16FP, and iSBC MM32FP modules can be stacked on the CPU baseboard in any combination.

Data Bus Structure

The MM-Series memory modules use a 32-bit wide data path with storage for byte parity that can accommodate 8-bit byte, 16-bit or 32-bit word data transfers. In addition, the data path is capable of independent byte operations. This means that one byte can be written while the other three bytes (or any other combination) can be read.

Parity

One parity bit is provided for each of the four, 8-bit bytes in the 32-bit wide data path. For special applications, the parity bits can serve as data bits making possible 9-, 18-, or 36-bit data transfers.

Memory Function

The module protocol supports standard dynamic RAM READ, WRITE, RAS* only REFRESH cycles, and CAS* before RAS* REFRESH.

Installation

The iSBC MM-Series memory modules are easily installed by the user. Each module includes all necessary connectors, screws, and other hardware for installation, either as a second stacked module or as a replacement for a module with less memory.

SPECIFICATIONS

Word Size Supported

8-, 16-, or 32-bits

Memory Size

iSBC MM01FP	1,048,576 bytes
iSBC MM02FP	2,097,152 bytes
iSBC MM04FP	4,194,304 bytes
iSBC MM08FP	8,388,608 bytes

Access Time (All Densities)

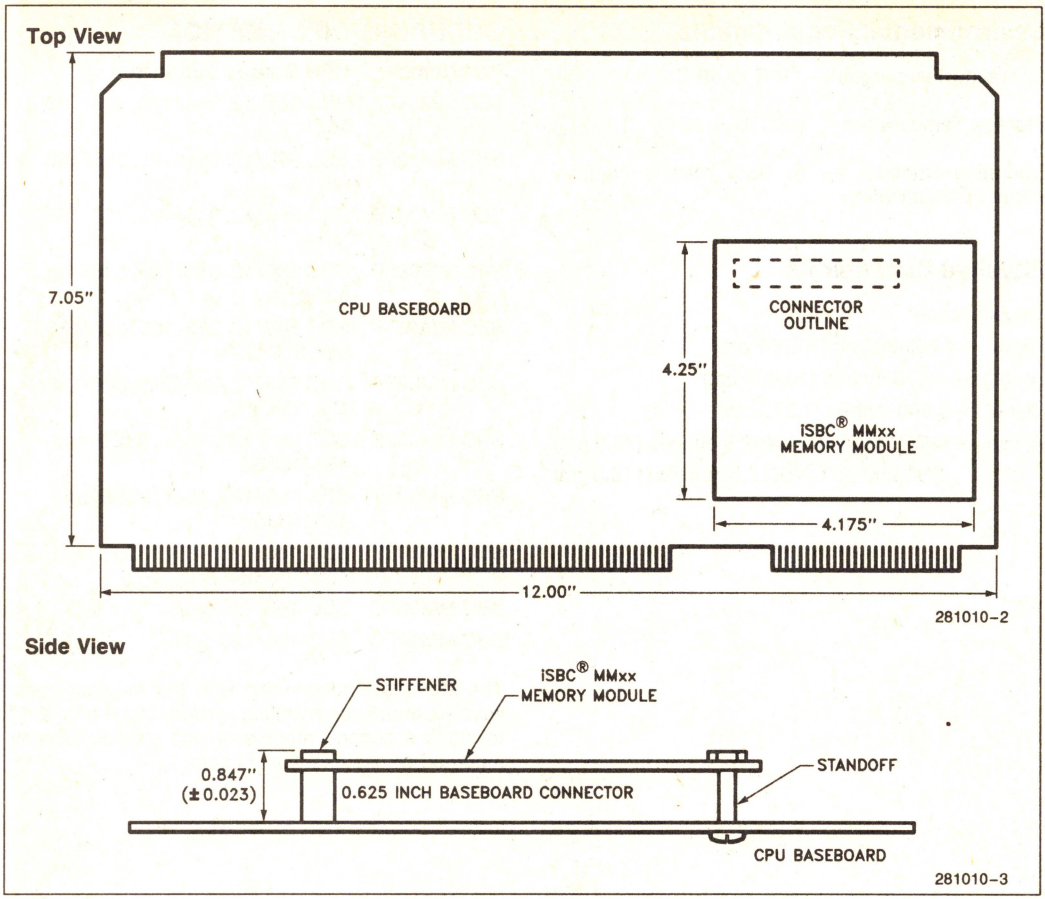
Read/Write — 107 ns (max)-MM0X

Read/Write — 88 ns (max)-MM0XFP

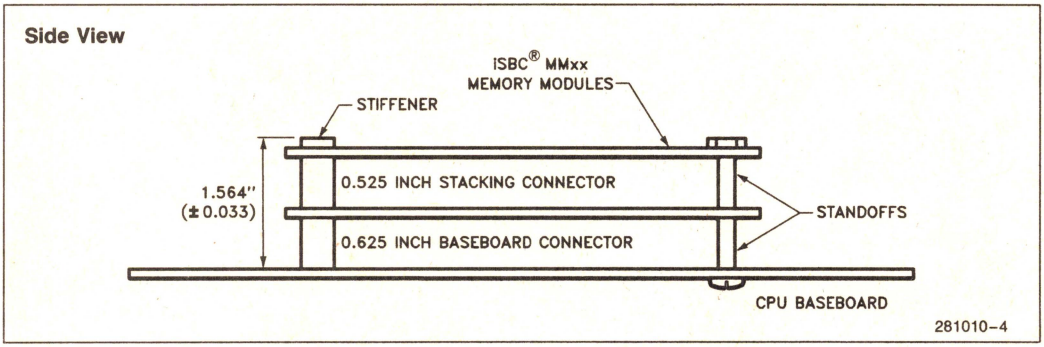
Power Requirements

Voltage −5 VDC ±5%

Memory addressing for the iSBC MM-Series memory modules is controlled by the host CPU board over the memory module interface. The maximum system RAM size is 64 MBytes.



Single iSBC MMxx Memory Module



Stacked iSBC MMxx Memory Modules

Environmental Requirements

Operating Temperature — 0°C to 60°C

Storage Temperature — 40°C to +75°C

Operating Humidity — To 95% relative humidity without condensation

Physical Dimensions

Module Alone:

Width — 4.250 inches (10,795 cm)

Length — 4.175 inches (10,604 cm)

Height — 0.500 inches (1,270 cm)

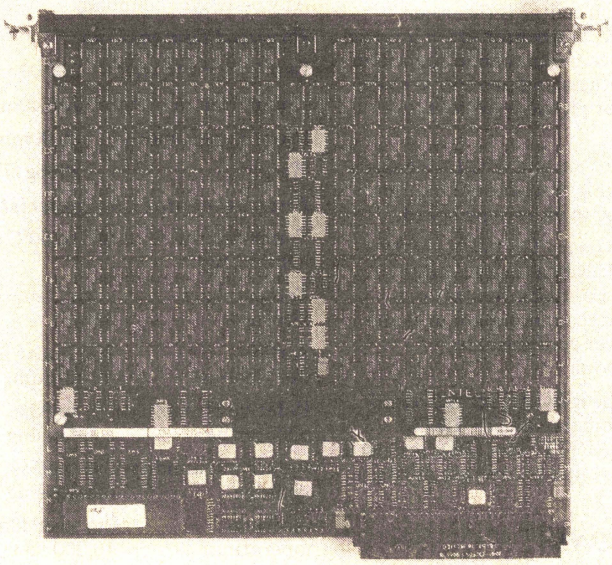
Weight — iSBC MM01/MM04: 2.5 ounces (70.0 gm)
iSBC MM02/MM08: 3.5 ounces (110.0 gm)

ORDERING INFORMATION

Part Number	CPU Boards Supported
SBC MM01FP	SBC 386/12, 386/125, 386/133, Mix
SBC MM02FP	SBC 386/12, 386/125, 386/133, Mix
SBC MM04FP	SBC 386/12, 386/125, 386/133, Mix
SBC MM08FP	SBC 386/12, 386/125, 386/133, Mix
SBC MM16FP	SBC 386/12, 386/125, 386/133, Mix, 386/258
SBC MM32FP	SBC 386/12, 386/125, 386/133, Mix, 386/258
SBC MM3008	SBC 486/125, 486/133SE, and 486/166SE
SBC MM3016	SBC 486/125, 486/133SE, and 486/166SE
SBC MM3032	SBC 486/125, 486/133SE, and 486/166SE
SBC MM08FC	SBC 486/125 only
SBC MM32FC	SBC 486/125 only

The Memory Modules ship with the required hardware (connectors, mounting screws, stand-offs, etc.) to stack a second module on the module already mounted on the base CPU board.

iSBC MEM/128 MEMORY BOARD



281145-1

The iSBC MEM/128 is Intel's highest capacity Multibus II Memory Board, containing 128 MBytes of DRAM. Designed for memory-intensive applications, the iSBC MEM/128 features a single PSB wait-state read and write access time, byte parity error detection, and requires only one slot in a Multibus II system. Board configuration is accomplished via interconnect space over the PSB bus.

- 128 MBytes of Dynamic RAM (DRAM)
- 32-bit PSB Bus Replier Interface
- 200 ns Read Access Time
- Byte Parity Error Detection
- Interconnect Space Support for Identification, Configuration, and Diagnostics
- Standard Eurocard Form Factor
- Occupies Only One Slot
- All Control Signals, Power and Ground supplied via the PSB 96-pin DIN Connector (P1)
- Initialization Checks run automatically at Power-up

iSBC MEM/128 MEMORY BOARD

PHYSICAL DESCRIPTION

The iSBC MEM/128 board is two boards joined together: the mother board which contains the bus interface and 64 MBytes of DRAM, and the daughter board which contains the other 64 MBytes of DRAM. The boards are connected with a 100-pin surface mount connector which uses gold-plated contacts and pads. Five nylon standoffs are used for physical support and reliability.

Functionally, the mother and daughter boards operate as a single unit. Because of this, they are typically referred to as a single board.

The iSBC MEM/128 board fits into a single 0.8" card slot in a Multibus II-compatible cardcage or system. Board configuration is handled through interconnect space over the PSB; there are no user-configurable components or jumpers.

The front panel of the iSBC MEM/128 board contains two light emitting diodes (LEDs) labeled TEST and PARITY. The LEDs are used for diagnostic support and to indicate parity errors.

SPECIFICATIONS

Word Sizes Supported

8, 16, 24, and 32-bits

Memory Size

128 MBytes of DRAM

Access Times (Multibus II PSB @ 10 MHz)

Read Operation	200 ns (without refresh cycle)
	400 ns (with refresh cycles)
Write Operation	100 ns from requester ready

Starting (Base) Address

Any 64 KByte boundary

Ending Address

Any 64 KByte boundary

Physical Dimensions

Depth (front to back)	220.00mm (8.66")
Height (top to bottom)	233.40mm (9.18")
Front Panel Width	20.00mm (0.79")
Weight	2036g (71 oz)

Environmental Characteristics

Temperature (Operating)	0°C to 50°C (32°F to 131°F)
Temperature (Storage)	−40°C to +70°C (−40°F to 158°F)
Airflow (for Cooling)	200 linear feet per minute (minimum)
Relative Humidity (Operating)	0% to 85%, non-condensing
Relative Humidity (storage)	0% to 95%, non-condensing
Altitude (operating)	0m to 3048m (0 feet to 10,000 feet)
Altitude (storage)	0m to 15,240m (0 feet to 50,000 feet)

Electrical Characteristics

Voltage	5V DC ±5%
Current	
Typical	6.25A (@80% of worst case)
Maximum	7.80A
Wattage	
Typical	31.25W (at nominal voltage)
Maximum	39.00W (at nominal voltage)

Multibus II I/O Products and MIX Boards

3

MIX ARCHITECTURE

Intel's Modular Interface eXtension (MIX) architecture is an easily customizable I/O solution that saves development time and costs. The MIX architecture is flexible enough to offer a wide range of I/O options, modular enough to be able to track the CPU and I/O technologies independently, and open enough to allow designers to build unique I/O solutions without having to re-engineer the CPU portion of the solution.

The MIX architecture offers the benefits and advantages of:

- ***Scalable price/performance for optimization of I/O throughput***
  Using the MIX architecture, you can implement a wide range of solutions—from high performance I/O controllers (baseboard and one MIX I/O module) to I/O server subsystems (baseboard and three MIX I/O modules). The selection of baseboards and I/O modules available allows you to create powerful I/O subsystems to fit your application. You can later reconfigure the same building blocks to fulfill new application requirements.
- ***Low-risk, quick time-to-market for development of custom I/O***
  Because you're working with a non-proprietary open standard with powerful development tools, design time is shortened. In addition, because the baseboard has already been developed by Intel, you can focus on the design tasks unique to your custom I/O. Therefore, the effort to implement custom I/O in a MULTIBUS II system is greatly reduced.
- ***Cost-effective upgradability for incorporation of new CPU or I/O technology***
  As new MIX baseboards and I/O modules are developed, they can be implemented in your design without abandoning the rest of a MIX I/O solution.

MIX ARCHITECTURE FEATURES

- Full compatibility with the MULTIBUS II (IEEE/ANSI 1296) Systems Architecture (MSA)
- Support for stacking of 1, 2, or 3 modules on the MIX baseboard. A MIX baseboard with a single MIX module fits into a single MULTIBUS II card slot
- MIX bus data width of 32, 16, or 8 bits
- Partitioning of I/O SBC architecture into a CPU and MULTIBUS II core and I/O module
- Multimaster bus ownership for support of master or slave MIX I/O modules
- Support for like or unlike MIX module stacking with dynamic Built-In Self Test (BIST) and interconnect capabilities
- Complete documentation available for building MIX I/O modules

October 1993
Order Number: 281009-004

MIX ARCHITECTURE

MULTIBUS II COMPATIBILITY

The MULTIBUS II Systems Architecture is optimized for efficiently interconnecting multiple intelligent microprocessor-based subsystems. The MULTIBUS II Parallel System Bus (PSB) provides the main communication backbone for the total system, while other elements of the architecture solve the system integration issues such as initialization, diagnostics, and standardized subsystem to subsystem signaling and data transfer. The MULTIBUS II architecture also accommodates many types of local bus extensions for solving local communication requirements within subsystems.

The MIX architecture adds a subsystem bus technology that provides a solution for high performance I/O and brings the capability of a high performance I/O server subsystem to the MULTIBUS II architecture.

PHYSICAL DECOUPLING OF CPU FROM I/O

MIX uses a baseboard plus modules approach to physically decouple the CPU technology of the baseboard from the I/O technology of the module. This decoupling has two benefits for I/O design. First, it allows evolving CPU and I/O technologies to be independently and easily incorporated into designs. Second, it allows a baseboard to change personality by adding or substituting I/O modules. This provides I/O flexibility while preserving your software investment.

CONNECTING CPU AND I/O VIA A HIGH PERFORMANCE BUS INTERFACE

The MIX architecture provides the high performance bus interface for coupling the CPU baseboard with the I/O modules. Elements of the MIX architecture and bus interface include:

Signal Set

The MIX bus consists of 130 signal, power, and ground connections. There are two types of signals: dedicated and bussed. Dedicated signals belong to specific modules in the MIX stack, while bussed signals are shared by all modules.

Address Capability

The MIX bus supports the full 4 GB physical addressing capability of Intel486™ microprocessors and other compatible microprocessors.

Data Paths

MIX supports 8, 16, and 32 bit physical data paths on MIX modules. The MIX baseboard data path is 32 bits.

MIX Bus Transfers

The baseboard can perform memory, I/O, and DMA transfers on the MIX bus. The baseboard can also perform a bus vectored interrupt transfer cycle. Bus master modules can perform memory transfers with the baseboard memory.

Arbitration

The MIX bus uses a simple round robin arbitration scheme between the baseboard and master modules to insure that all modules and the baseboard have guaranteed access times to shared baseboard memory and have a guaranteed percentage of the shared memory bandwidth.

Interrupts

Each MIX module has one dedicated interrupt line. Each module also has an option line that can be used as an interrupt line.

DMA

The MIX bus supports DMA transfers between modules and the baseboard memory. Both single-cycle (fly-by) and two-cycle DMA transfers are supported.

Configuration Support

MIX configuration support has been designed to be compatible with the MULTIBUS II interconnect space architecture. MIX modules are viewed as baseboard functions by agents on the parallel system bus. The baseboard microcontroller reads the interconnect information stored in the EEPROM of each module present in the MIX stack to build the function record in baseboard interconnect space.

Built-In Self Test (BIST) Support

MIX provides the capability for BIST code resident in module EPROM to be downloaded and executed as an extension of the baseboard BIST.

MIX ARCHITECTURE

OPEN INTERFACE FOR I/O MODULE DEVELOPMENT

MIX provides an excellent platform for building MULTIBUS II I/O solutions. A complete set of manuals, design specifications and design examples for building MIX I/O modules is available from Intel.

MIX ARCHITECTURE SPECIFICATIONS

General

Bus Type:	Asynchronous
Theoretical Bandwidth:	22 MB/s
Typical Bandwidth:	10 MB/s
Bus Overhead:	7% Xchange/Refresh
Interrupt Sources:	Any Module
Bus Vector Support:	Yes
Arbitration Scheme:	Fairness (Rd-Robin)
Module Maximum:	3 (master or slave)
Length of Bus Hold:	8 microsec (typical)
Flag Byte Support:	Yes

Lines

130 total signal, power and ground lines:

Number	Functional Group
37	Address
32	Data
7	Transfer Control
6	Arbitration
3	Interrupt
6	DMA
3	Option
7	Configuration
9	+5 VDC
13	GND
2	+12 VDC
2	−12 VDC
3	**Reserved**

Baseboard Address Range

Memory	4 GB
I/O	64 KB

Module Address Range

Memory	256 MB
I/O	1 KB

Data Path

8, 16, and 32 bit

MIX Connector

Connector Type: Surface Mount

Connector Pads: 130

MIX Expansion Module

Module Height:	8.9 inches
Module Depth:	3.75 inches
Module Area:	33 square inches
Max Configuration:	1 Baseboard 3 Modules

MIX Bus Power Limits

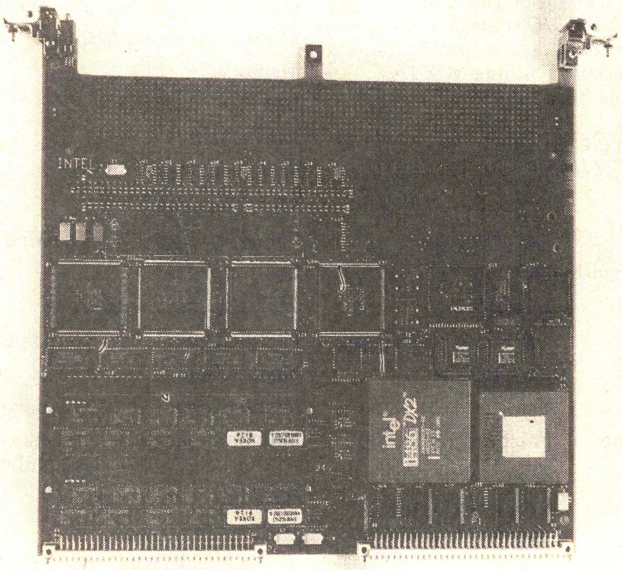
Voltage (VDC)	Total Current (Amps)
+5 (+5%, −2%)	9.0
+12 (+5%, −5%)	1.5
−12 (+5%, −5%)	1.5

Module Power Limits

Nominal Voltage (VDC)	Max Current per Module (Amps)
+5	3.0
+12	0.5
−12	0.5

Thermal Limit (all sources): 20 Watts max per module

MIX BASEBOARDS



281009-41

MIX 486/DX66, 486/DX33, AND 486/SX33 BASEBOARDS

The Intel Modular Interface eXtension (MIX) 486/DX66, 486/DX33, and 486/SX33 baseboards represent the leading edge in customizable mezzanine I/O solutions. These technically advanced baseboards are designed for cost-effective CPU and I/O technology upgrades, low-risk and quick time-to-market custom I/O development, and scalable price/performance I/O throughput optimization.

This family of Intel486™ microprocessor-based MIX baseboards is compatible with and has greater performance than previous Intel MIX baseboard families.

- **High throughput to handle the most demanding I/O applications.** The MIX baseboards are available with Intel486 DX2, DX, and SX CPUs operating at 66 MHz and 33 MHz for high-performance processing. The baseboard design combines high-powered Intel486 CPUs together with a 128KB secondary (L2) cache, a 33 MHz I/O bus, and a 16.5 advanced DMA (ADMA) controller for lightning-fast throughput to speed up your applications.
- **Flexible solutions for changing system requirements.** Not only can you select the right performance level CPU for your application, but you can also expand on-board DRAM up to 32 MB. Also, up to three MIX modules can be stacked on a baseboard. This means you can create powerful subsystems and then easily reconfigure them to meet new requirements.
- **Cost-effective upgradability.** This new family of baseboards is compatible with existing MIX baseboards, I/O modules, and software, saving you valuable development time and money when upgrading your system.

MIX BASEBOARDS

Intel486™ CPU-BASED BASEBOARD FAMILY FEATURES

- Intel486 SX and DX CPUs running at 33 MHz or an Intel486 DX2 running at 66 MHz
- 33 MHz bus
- 8 KB internal and 128 KB second level cache for instruction and data
- SIMM DRAM (two SIMM sockets) expandable to 32 MB using industry-standard low-profile SIMMs (70 ns)
- 82C258 Advanced DMA (ADMA) controller operating at 16.5 MHz
- Integrated DMA address generator logic (DAG compatible) for 32-bit address range
- Two 8259 programmable interrupt controllers (PICs) for 14 programmable interrupt levels
- 82C54 programmable interval timer (PIT) with three event counters/timers
- 82389 Message Passing Coprocessor
- 87C51 (Interconnect) Microcontroller
- MULTIBUS II Parallel System Bus Interface

High-Speed Intel486™ Microprocessor Core

These baseboards obtain their I/O processing power from an Intel486 DX2 microprocessor operating at 66 MHz or an Intel486 DX or SX microprocessor operating at 33 MHz. The baseboards contain 8 KB of internal cache and use 128 KB (with parity) of secondary (L2) cache for instructions and data.

The Protected Virtual Address Mode (PVAM) of the microprocessor provides the baseboards with a full four gigabytes of addressability. The top gigabyte is used for baseboard Flash/EPROM and MIX Module memory access. The lower three gigabytes are divided between baseboard DRAM and Parallel System Bus access. PVAM operation also provides support for code protection, virtual memory, and paging mechanisms.

In addition, the microprocessor has a self-test capability which is utilized in the board's power up BIST testing. This function can be disabled via a board jumper option.

Expandable Memory

A 2-way interleaved SIMM configuration provides a high-performance i486 bus memory subsystem.

The baseboards are supplied with either 0 MB or 8 MB of baseboard memory, and can accept up to 32 MB of DRAM in the form of low-profile tin-socketed 70 ns SIMMs. Memory configurations of 2, 4, 8, 16, and 32 MB are supported.

Baseboard DRAM is directly accessible to bus masters on the MIX bus. Byte parity protection is used for DRAM error checking on the board.

Improved DMA Performance for Fast Memory Transfers

The baseboards use the 82C258 ADMA controller operating at 16.5 MHz, DMA Address Generator (DAG) gate array for 32-bit, 4 gigabyte addressing, and a fast page mode DRAM control logic to handle high speed data transfers between baseboard DRAM and both the MPC and MIX modules.

The ADMA controller provides four independent channels for DMA service. Channels 0 and 1 are used to support the MIX stack; these two channels can be programmed to support any of the 3 MIX module positions. Channels 2 and 3 are used to service the Message Passing Coprocessor (MPC). All four ADMA channels can access memory on any MIX module by programming the channel's paging and control registers in the DAG.

Three transfer modes are supported: burst, single-cycle (fly-by), and two-cycle. To the MIX bus, burst mode transfers look like single-cycle transfers. Burst mode transfers from the baseboard MPC to the DRAM peak at 27 MB/s, and burst mode transfers from the baseboard DRAM to MIX modules have a maximum transfer rate of 17 MB/s.

MULTIBUS II Systems Architecture Support

These baseboards utilize the 82389 Message Passing Coprocessor (MPC) to provide a full 32-bit interface to the MULTIBUS II Parallel System Bus. Firmware is also provided that contains baseboard BIST and Initialization and IDX code. Also included is the capability to download BIST code from MIX module EPROM and execute the code to test the modules in the MIX stack.

MIX BASEBOARDS

A High-Speed Engine for I/O Processing

The baseboards support up to three MIX modules, one of which can be a full-size module. The MIX bus supports 32, 16, and 8-bit data transfers and allows MIX modules to be either masters or slaves on the MIX bus.

A custom gate array device on the baseboard controls the baseboard interface to the MIX bus and implements the MIX bus arbitration logic. The gate array implements the standard MIX round-robin arbitration algorithm which provides guaranteed access to the MIX bus by the baseboard and module masters. The capability to modify certain arbitration parameters is also provided.

Compatibility

The Intel486 CPU-based family of baseboards is software compatible with existing baseboards. Software developed for the MIX386/020A or the MIX 486/020A baseboards will run on the new baseboards. You may, however, wish to optimize some of your existing software to take advantage of the new features and increased access speed of the new baseboard.

MIX BASEBOARD SPECIFICATIONS

BASEBOARD SPECIFICATIONS

Baseboard	**CPU**
MIX 486/SX33	Intel486 SX CPU
MIX 486/DX33	Intel486 DX CPU
MIX 486/DX66	Intel486 DX2 CPU

Clock Rates

CPU	33/66 MHz
82C258 ADMA	16.5 MHz
87C51 Microcontroller	12 MHz
82C54 Timer	(Programmable)

EPROM/FLASH MEMORY

Two 32-pin sites

DRAM MEMORY

The baseboards can be ordered with 0 MB or 8 MB of SIMM memory. Memory may be increased up to 32 MB with low-profile, tin-socketed 70 ns SIMMs.

INTERRUPT CAPABILITIES

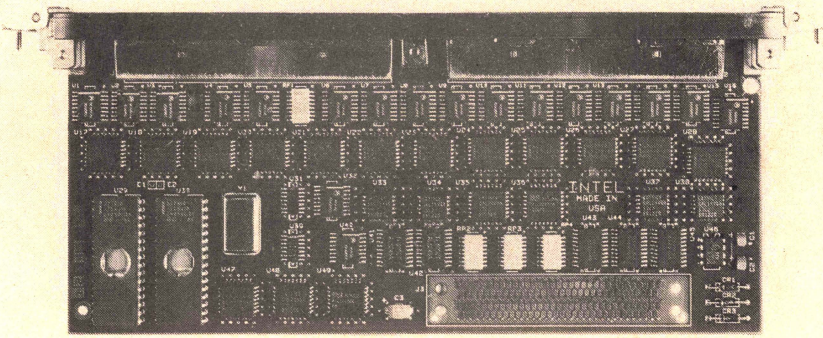
14 programmable interrupt levels

INTERFACES

- P1, full PSB
- P2, power only
- MIX

PHYSICAL CHARACTERISTICS

Standard MULTIBUS II board



281009-42

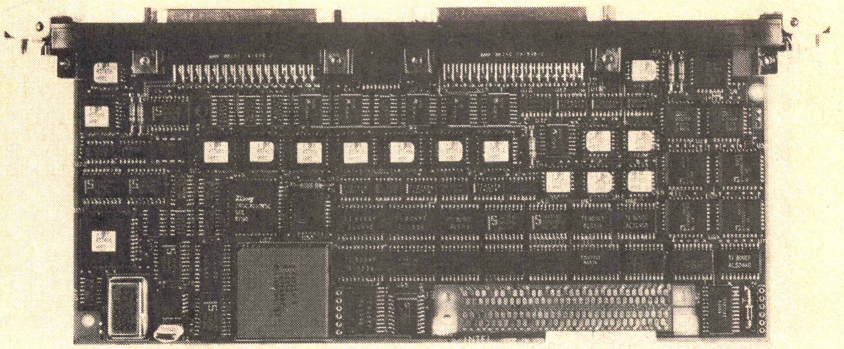
MIX 450 ASYNCHRONOUS TERMINAL CONTROLLER

The MIX 450 terminal module, when combined with a MIX baseboard, provides high performance terminal server capability for MULTIBUS II systems. The MIX 450 module, as a single module on the MIX baseboard, is a powerful 12 port terminal I/O controller. The module can also be stacked three high to expand the terminal support to 36 ports. Stacking the MIX 450 with other MIX modules allows the system designer to build to a multi-function I/O server with terminal capabilities.

FEATURES

- 12 ports per board, RS232C compatible
- 13 signal support, RJ45 (Phone Jack Style) shielded connectors
- Asynchronous Terminal Control Software (ATCS) for interrupt processing, character handling and modem support
- Multiple host support including dynamic line switching
- Resident firmware to support BIST

MIX I/O MODULES



281009-43

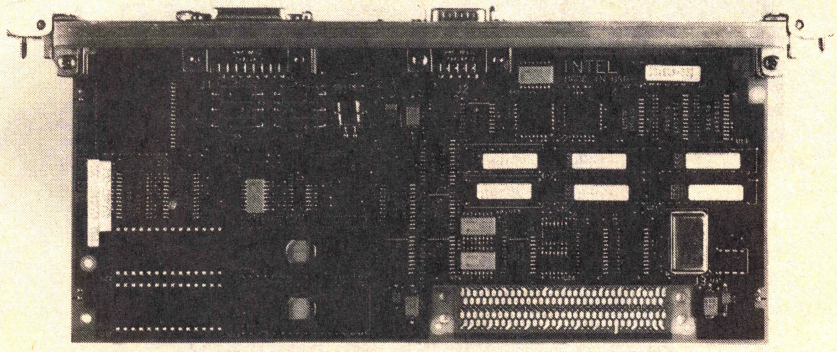
MIX 420 WIDE AREA NETWORK CONTROLLER

The MIX 420 module combines two high speed synchronous channels with a MIX baseboard to build a high performance Wide Area Network (WAN) platform. The MIX 420 modules can be stacked to a maximum level of three modules for expansion up to six channels. The MIX 420 can be stacked with other MIX modules, allowing the system designer to build a multi-function I/O server with WAN capabilities.

FEATURES

- Two Independent High Speed Synchronous Channels using a 10 MHz 85C30 Serial Communication Controller (SCC)
- High Performance communications capable of 256 kbit/second, with ADMA, and Bypass (slave) speed modes
- Flexible communications with an 82C54 Programmable Interface Timer and either channel interrupts or hardware interrupts using the 82C59 component
- "Smart Cable" interface using AT&T General Purpose Synchronous (GPSYNC) Standard
- Designed as a hardware platform for Synchronous Protocol
- Support, including SDLC/HDLC, SNA, Bisync/Async, SNA, X.25, X.21, X.21 BIS, LU6.2
- Firmware containing BIST code

MIX I/O MODULES



281009-44

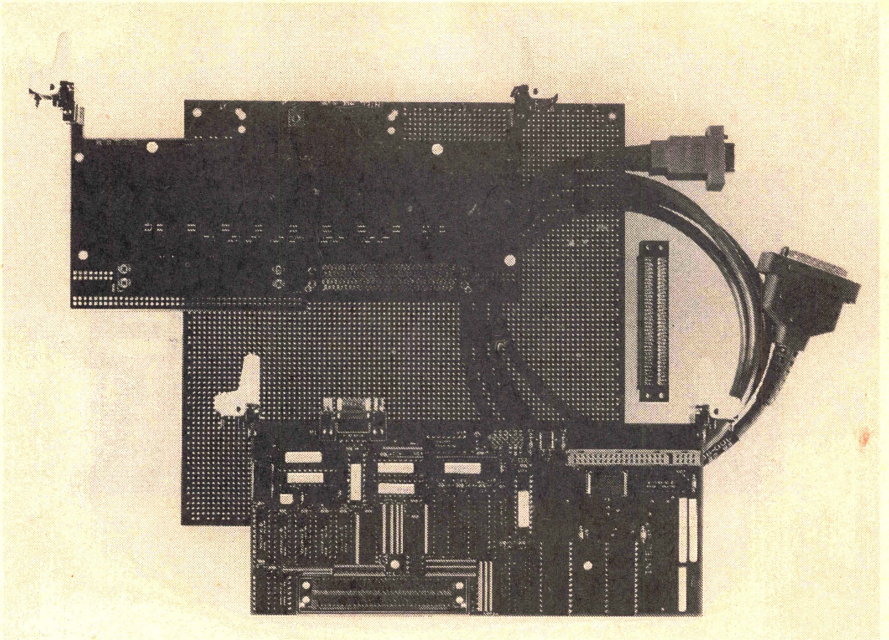
MIX 560 ETHERNET CONTROLLER

The MIX 560 Ethernet Module combines an 82586 LAN Coprocessor, 82501 Ethernet Serial Interface, and 64 Kbytes of high speed SRAM data buffer to provide high performance Ethernet MIX I/O capabilities. The MIX 560 can be used either in a MIX module stack, to provide Ethernet capabilities to a MULTIBUS II MIX I/O server subsystem, or as a single module on the MIX baseboard, to provide a high performance MIX-based Ethernet controller. Stacking the MIX 560 Ethernet module with other MIX I/O modules allows the system designer to manage the system I/O requirements with a tailored MULTIBUS II I/O subsystem that includes Ethernet communications.

FEATURES

- 82586 LAN Coprocessor operating at 10 MHz
- 82501 Ethernet Serial Interface
- 64 Kbytes of SRAM data buffer for handling communications from the MIX baseboard to the Ethernet Interface
- Support for 128K–265K EPROM
- Firmware containing BIST code
- LED for 82586 Activity
- Serial interface for system console or debug

MIX I/O MODULES



281009-45

MIX DEVELOPMENT MODULES

A requirement for many MULTIBUS II designs is the incorporation of special I/O. This is I/O that may be unique to the application or require an I/O controller not readily available in the market. To date, such a requirement typically would have necessitated the non-trivial task of designing a dedicated MULTIBUS II I/O controller. With the MIX Development Modules, the task of developing a MULTIBUS II I/O solution is simplified. The MIX bus is a straight-forward, well documented interface for developing MIX I/O modules. In addition, the MIX Development Modules provide you with all the hardware and documentation needed to easily develop your own MIX I/O module that stacks on the MIX baseboard. This combination greatly simplifies the task of developing your complete I/O solution for MULTIBUS II systems.

FEATURES

- MIXMOD1 Test Module—used for testing MIX module hardware and software designs
- MIXMOD2 Breadboard Module—used for wire-wrapping and building a prototype module design
- MIXMOD3 Debug Module—used to mount a MIX module with its component side up
- Complete documentation set providing all the information needed to develop a MIX module
- Optional Intel Field Systems Engineer consulting to help you better focus your development team and save development time

MIX MODULE SPECIFICATION SUMMARY

MIX 450 ASYNCHRONOUS TERMINAL CONTROLLER

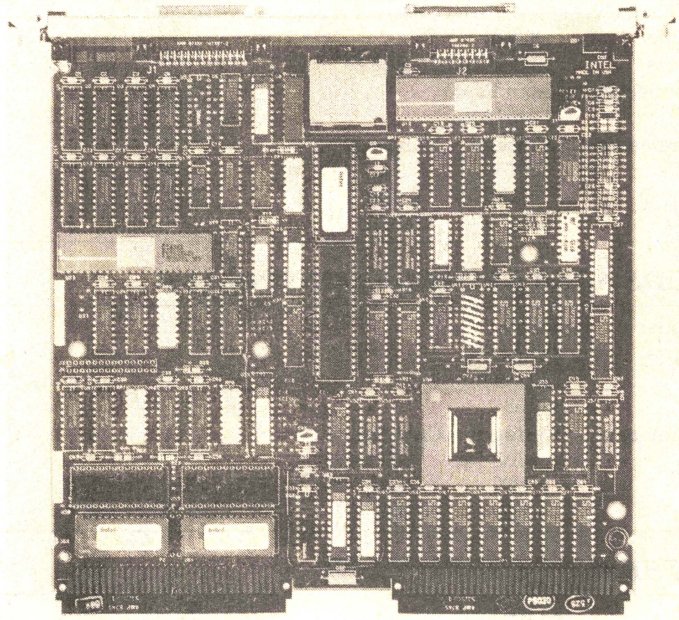
Serial Controller	82510	
Serial I/O Ports	12 per module RS232C	
Serial I/O Connectors	8-pin RJ-45	
Baud Rate	110–19.2K	
Power Requirements	**Voltage**	**Current**
	+5V	3A
	+12V	0.02A
	−12V	0.02A

MIX 420 WIDE AREA NETWORK CONTROLLER

Serial Controller	85C30	
Serial I/O Ports	2 channels	
Baud Rate	110–64K	
Timer	82C54	
Power Requirements	**Voltage**	**Current**
	+5V	3A
	+12V	0.02A
	−12V	0.02A

MIX 560 ETHERNET CONTROLLER

SRAM	64 KB	
EPROM	Two 32-pin JEDEC sites	
Ethernet	One channel, 15-pin connector	
Controller	82586 LAN Controller	
Serial I/O	One channel RS232C, 9-pin connector	
Controller	82510	
Support	IBM compatible	
Interrupts	14	
Power Requirements	**Voltage**	**Current**
	+5V	3A
	+12V	0.02A
	−12V	0.02A



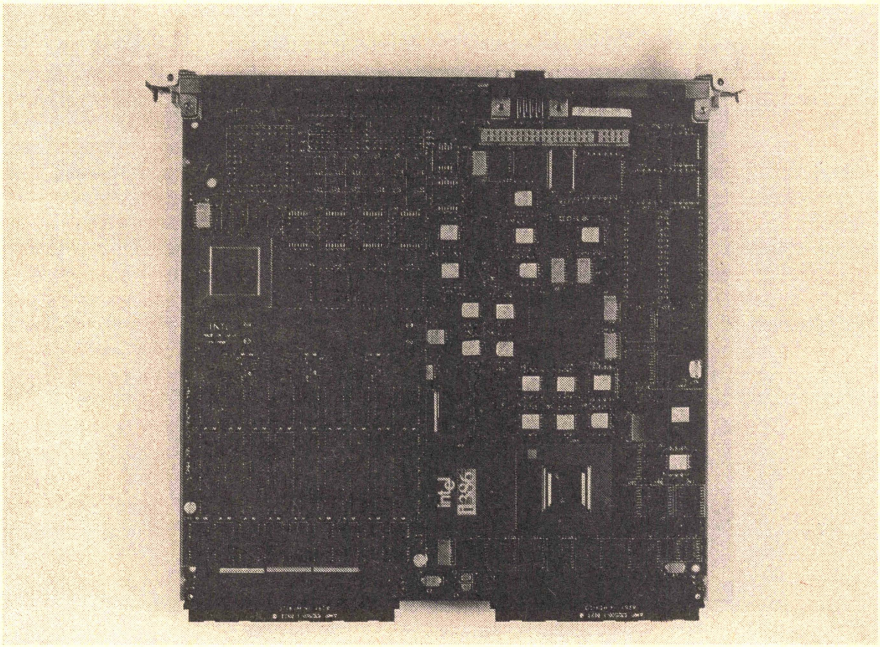
281009-46

iSBC 186/450 ASYNCHRONOUS TERMINAL CONTROLLER

The iSBC 186/450 is a high performance intelligent terminal controller. The 80C186 CPU, 512K RAM and ATCS served software allow this board to off load the terminal I/O processing from the MULTIBUS II application CPUs.

FEATURES

- 80C186-based microprocessor operating at 12.5 MHz
- 512K RAM, 128K or 256K EPROM
- Full MULTIBUS II Parallel System Bus interface provided by the Message Passing Coprocessor (MPC)
- 12 ports, RS232C compatible
- 8 signal support, RJ45 (Phone Jack Style) shielded connectors
- Asynchronous Terminal Control Software (ATCS) for interrupt processing, character handling and modem support
- Multiple host support including dynamic line switching
- Resident firmware to support BIST, host-to-controller software download



281009-47

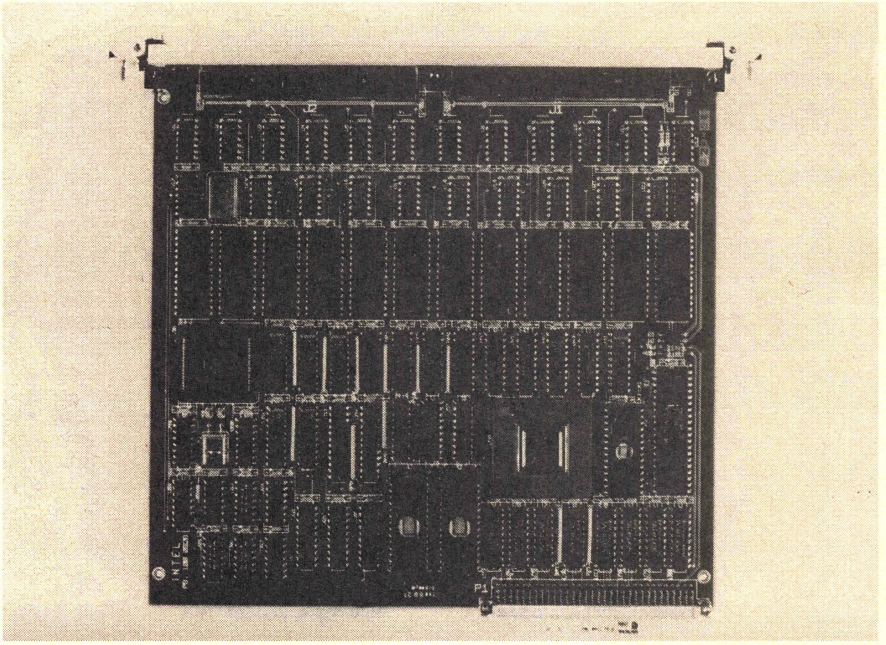
iSBC 186/410 WIDE AREA NETWORK CONTROLLER

The iSBC 186/410 MULTIBUS II Serial Communications Board is an intelligent 6-channel communications processor that addresses the needs of many standard communication applications. The board brings flexibility to the application with its multiple serial channels as well as I/O expansion through the SBX connections.

The iSBC 186/410 is designed to support serial communication within the system. The iSBC 186/410 board supports asynchronous, byte synchronous, and bit-synchronous (HDLC/SDLC) communications on the two full/half duplex RS232C or RS422A channels. On the remaining four channels, only asynchronous mode (RS232C) is supported in either full or half duplex operation. Each serial channel can be individually programmed for different baud rates to allow system configurations with differing terminal types.

FEATURES

- 8 MHz 80C186 microprocessor
- Six serial communication channels, two RS232C or RS422A, four RS232C only, front panel connections
- DMA controller provides 4 independent DMA channels
- 512 KB DRAM provided, four 28-pin JEDEC sites available for EPROM
- Two iSBX connectors provided for I/O expansion



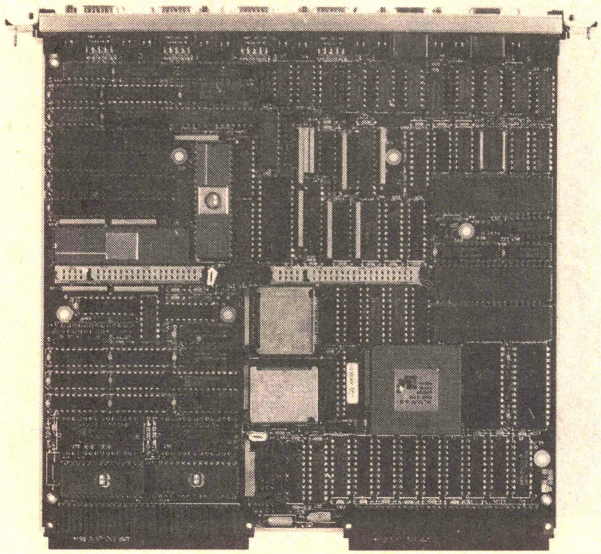
281009-48

iSBC 186/530 ETHERNET CONTROLLER

The iSBC 186/530 Multibus II Ethernet Controller is a dedicated IEEE 802.3 compatible front-end processor. The board's 8 MHz 80186, 512K DRAM, and host-to-controller software download capability allows the board to off-load LAN communications functions and I/O software processing from one or all of a Multibus II system's host CPU boards.

FEATURES

- Provides IEEE 802.3/Ethernet compatible networking capability for MULTIBUS II systems
- Resident firmware to support BIST, IDX, and host-to-controller software download
- Four 28-pin JEDEC sites, expandable to 8 sites with iSBC 341 MULTIMODULE for a maximum of 512 KB of EPROM
- One RS232C serial port for use in debug and testing
- MULTIBUS II Parallel System Bus interface with full message passing capability



281009-49

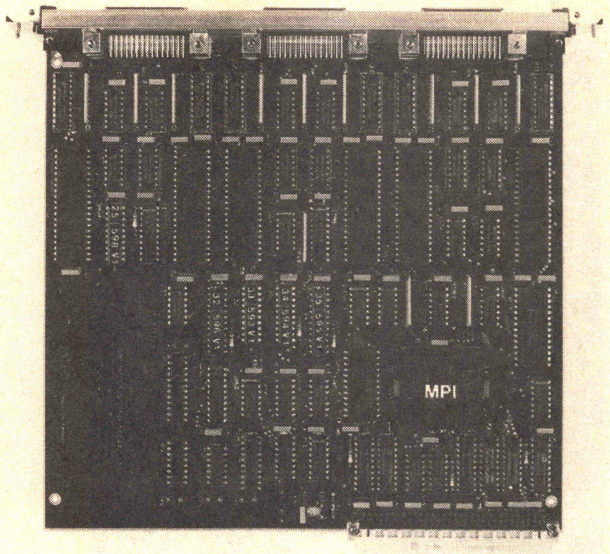
iSBC 386/258 SCSI PERIPHERAL CONTROLLER

The iSBC 386/258 is a high-performance peripheral controller that combines powerful I/O performance and access to SCSI peripherals for MULTIBUS II applications.

Minicomputer-level I/O performance is achieved by utilizing the Intel386 microprocessor and a large data cache. The added power of the Intel386 processor gives the iSBC 386/258 the potential of off-load tasks from other system CPUs as an I/O server. The SCSI standard has achieved wide acceptance because of its extensive capabilities and excellent performance.

FEATURES

- 16 MHz Intel386 microprocessor
- CSM002 module support
- Common Command Set (CCS) SCSI peripheral support
- Asynchronous SCSI to 1.5 MB/s, synchronous to 4.0 MB/s
- Two versions: single ended SCSI port only or dual SCSI ports
- Firmware support for BIST, IDX, slave test handler, and downloader
- Peripheral Communications Interface (258-PCI) firmware
- Two MSA EPROMs are included in the shipping container



281009-50

iSBC MPI/519 72 CHANNEL DIGITAL I/O BOARD

The iSBC MPI/519 is a digital I/O interface board which provides 72 parallel channels of TTL level I/O in Multibus II I/O space. The board suitable for applications such as industrial automation, printer interface, or for low cost inter-chassis communications requiring multiple parallel I/O lines. It is capable of receiving interrupts from other Multibus II agents, as well as generating interrupts from up to 8 sources.

iSBC MPI/519 FEATURES

- 72 channels of TTL level I/O in banks channels each
- Banks configurable for general purpose industrial I/O or as Centronics compatible ports
- Socketed buffer drivers and resistor networks for configuring I/O as active high or active low
- 8 interrupt request lines, one can be configured for broadcast interrupt.
- Precision interval pulse triggering on three I/O lines, one on each of three banks
- Jumperless configuration

I/O BOARD SPECIFICATION SUMMARY

ISBC 186/450 TERMINAL CONTROLLER

CPU	80186 @12.5 MHz	
Serial Controller	82510	
RAM	512 KB dynamic (32 KB dual ported)	
EPROM/ROM	Two 32-pin sites	
Serial I/O Ports	12 asynchronous RS232C, 8-pin RJ45	
Baud Rate	110–19.2K	
Interrupts	5 levels	
Power Requirements	**Voltage**	**Current**
	+5V	3A
	+12V	0.02A
	−12V	0.02A

iSBC 186/410 WIDE AREA NETWORK CONTROLLER

CPU	80186 @ 8 MHz	
Serial Controller	82530	
DRAM	512 KB	
EPROM/ROM	256 KB (27512), 512 KB with iSBC 341 board	
Serial I/O Ports	6 total (expandable to 10 with iSBC 354 board	
Timers	3	
Interrupts	16 levels, 15 sources	
Power Requirements	**Voltage**	**Current**
	+5V	8.22A
	+12V	150 mA
	−12V	150 mA

iSBC 186/530 ETHERNET CONTROLLER

CPU	80186 @ 8 MHz	
DRAM	512 KB	
EPROM	Four 28-pin JEDEC sites, up to 256 KB (max.) using 27512 devices, up to 512 KB (total) iSBC 341 module	
Ethernet Controller	One channel, 15-pin connector 82586 LAN Controller	
Serial I/O Controller Leads Supported	One channel RS232C, 15-pin connector 8031 TD, RD	
Timers	3	
Interrupts	5 levels with 5 on-board sources and 255 sources for PSB Bus	
Power Requirements	**Voltage**	**Current**
	+5V	8.8A
	+12V	0.05A
	−12V	0.05A

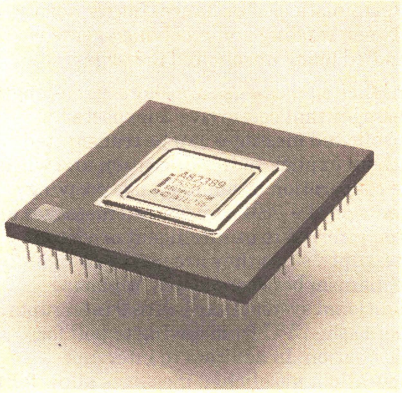
I/O BOARD SPECIFICATION SUMMARY

iSBC 386/258 PERIPHERAL CONTROLLER

CPU	i386 @ 16 MHz
DMA	82258 @ 8 MHz
DRAM	1 or 4 MB
EPROM	Two 32-pin sites
SCSI	ANSI X3.131-1986. Two versions available: single or dual (single and differential ends).
Device Support	Up to 7 other adapters, up to 56 devices
Transfer Rate	4 MB/sec
Serial I/O Ports	1 RS232C
Timers/Counters	3 16-bit
iSBX (IEEE P959)	1 8-/16-bit
Backplane Connectors	P1-PSBg P2-SCSI
Compatible Drives:	
Hard Disks	Maxtor, Micropolis, Hewlett-Packard, Quantum Pro, Siemens
Tape Drive	Archive
Adapters	Adaptec (SCSI to ESDI), ADP (NCR floppy)
Power Requirements	**Voltage** **Current**
	+5V 11A
	+12V 0.05A
	−12V 0.05A

iSBC MPI/519 72 CHANNEL DIGITAL I/O BOARD

Standard Multibus II board		
Backplane Connectors	P1-Slave PSB	
	Centronics	
	Each of the three I/O banks can be used as a Centronics compatible interface. Bank 1 is configurable to minimize handshaking and bus transactions when used as a printer interface.	
I/O connector	3 Positronics ODD44F500TX	
I/O Buffer and Resistors Supplied	**Bidirectional Buffers**	**Unidirectional Buffers**
	74ALS645	74ALS09
	Resistor Networks	
	2.2 KΩ	
Other Components Supported	**Bidirectional Buffers**	**Unidirectional Buffers**
	74ALS638-74AS638	74ALS00-74AS00
	74ALS639-74AS639	74ALS08-74AS08
	74ALS640-74AS640	74LS32-74AS32
	74ALS643-74AS643	74LS37-74AS37
		74LS38-74AS38
	or equivalent	or equivalent
	Resistor Networks	
	9- or 10-pin SIPs, all values supported	
Power Requirements	**Voltage**	**Current**
	+5V ±5V	3A



281009-6

82389—MULTIBUS II MESSAGE PASSING COPROCESSOR

The 82389 MPC is a highly integrated VLSI CMOS device that maximizes the performance of a Multibus II based multiprocessor system. The MPC implements the full message passing protocol as well as the functions (arbitration, transfer and exception cycle protocols) of the PSB bus interface control as defined in the IEEE/ANSI Standard 1296.

The 82389 MPC is designed to interface with an 32-, 16- or 8-bit processor. It provides support for message passing, interconnect space, memory, and I/O references on the PSB. In addition, the 82389 MPC component is designed to simplify implementation of dual port memory functions for those designs which will co-exist with the message passing communications protocol.

MPC FEATURES

- Single Chip Interface for the Parallel System Bus
- 1.0u CMOS Technology
- 149-pin Ceramic PGA Package (15 × 15 Grid)
- Optimized for Real-Time Response (Maximum 900 ns for 32-byte Interrupt Packet)
- Processor Independent Interface to the Parallel System Bus
- Supports co-existence of dual port and message passing communication protocols
- Dual Buffer Input and Output DMA capabilities

MPC 82389 INTERFACES

The three primary interfaces to the MPC (PSB, Interface Host, CPU Interface and Interconnect Interface) all function asynchronously to one another. This is accomplished through the use of internal latches and FIFOs that allow references to occur simultaneously on all interfaces. In addition to the three primary interfaces, the MPC contains a Dual-Port interface which provides compatibility with shared memory system implementations and software.

The PSB Interface

The PSB Interface is the synchronous, communications pathway in a Multibus II system. The PSB is a full 32-bit interface to other boards in the Multibus II chassis. The PSB interface supports PSB arbitration, data transfer and error handling.

The Host CPU Interface

The Host CPU Interface is a set of addressable registers and ports that is the private pathway for the local microprocessor on the Multibus II board. The Host CPU interface connects a 32-, 16- or 8-bit processor to the MPC. The Host CPU Interface supports direct references to memory, I/O, and interconnect address space on the PSB. The Host CPU Interface also supports DMA operations. The Multibus II PSB and the MPC are defined to be processor independent.

The Interconnect Interface

The Interconnect Interface provides a path for added board functionality that is independent from the host CPU. The Interconnect Interface is an 8-bit communication interface which requires the MPC to be connected to a microcontroller or a simple state machine. A microcontroller will perform tasks such as board configuration at start-up and local diagnostics. All interconnect bus signals are asynchronous to the bus clock and to the local bus signals.

The Interconnect space of an agent is the only required bus space by the IEEE/ANSI 1296 specification and has a 512-byte register range.

The Dual Port Interface

The Dual Port interface supports shared memory accesses between agents on the PSB. The MPC contains programmable address recognizers and PSB cycle control. In order to fully implement dual-port memory, some additional dual-port memory controller logic is required.

MAJOR MPC OPERATIONS

The MPC standardizes the signaling and data transfer between multiple intelligent agents within a Multibus II system. The traditional address spaces of memory and I/O were considered inadequate to accomplish this standardization task, so a new address space, called message space, was added. The movement of information in message space is called message passing. The MPC supports two types of messages: solicited and unsolicited.

Solicited messages are used to transfer large amounts of data. Up to 16 Mbytes (less 1 byte) of data can be transferred in a single solicited message transmission sequence. Solicited message transfers require the receiving agent to explicitly allocate a buffer. Data is packetized and reconstructed by the MPC to optimize PSB utilitzation and maintain deterministic performance. Buffer negotiation between sending and receiving agents is handled using unsolicited messages.

Unsolicited messages are short, fixed-length messages that can arrive unexpectedly. Unsolicited messages can be transmitted without explicit buffer allocation and without the cooperation of sending and receiving agents on the PSB. Unsolicited messages are often referred to as intelligent or virtual interrupts, since they are used as a signaling mechanism between boards, replacing traditional system (hard-wired) interrupts and freeing the CPU from having to poll for information. In addition to interrupt generation, unsolicited messages allow for up to 28 bytes of user data.

MPC Specifications

Operating Temperature . . . . . 0°C to +70°C

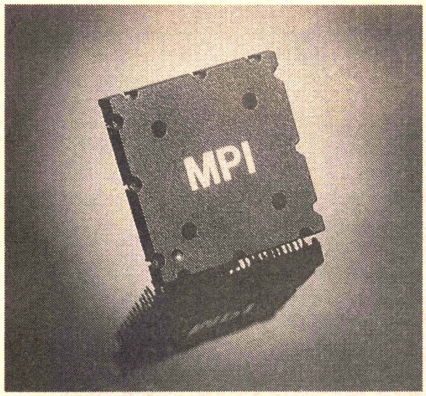
Storage Temperature . . . . . −65°C to +150°C

Voltage on Any Pin . . . . . 0.5V to V_{CC} +0.5V

Power Dissipation 2.5W

DC and AC Specifications are available in the 82389, Message Passing Coprocessor Datasheet.
(Intel Order Number 290145-003)

MULTIBUS II BUS INTERFACE SILICON PRODUCTS



281009-51

MPI—MULTIBUS II PERIPHERAL INTERFACE*

The MULTIBUS II Peripheral Interface, MPI, is a single chip, "replier only" Parallel System Bus interface device. The MPI implements a IEEE/ANSI 1296 Replier State Machine, seen in Figure 2. All error conditions are monitored and generated if appropriate.

MPI FEATURES

- Replier in I/O Space
  - — 2 Kbytes address on each board
  - — 8/16 bus data width agents
- No application CPU required
- No support microcontroller required
- Supports up to 8 local interrupt sources
- Sends/Receives broadcast messages
- Sends/Receives unsolicited messages (without data)
- Complete arbitration protocol
  - — Fair and High Priority modes are supported
- ANSI/IEEE 1296 compliant
- 124-pin plastic PGA package

DESCRIPTION

The MPI component is a 16-bit integrated CMOS interface component compliant with the IEEE/ANSI 1296 standard and is compatible with other board products using the 82389 Message Passing Coprocessor. It supports data transfer in I/O space, as defined by the IEEE Specification. It is particularly suited to the design of low cost, non-intelligent I/O boards. Since the MPI component incorporates all the interface logic, except for five high current buffer drivers, it simplifies and accelerates I/O board design. The local interface is designed to provide a simple interface to I/O board components. The MPI also includes configuration registers which are programmed from the PSB to suit a variety of applications.

Interconnect Space

The MPI component supports IEEE/ANSI 1296 compliant interconnect space. Its registers are only accessed via the Parallel System Bus. The MPI includes the interface logic to support an external local memory device or PAL to implement most of the interconnect registers. Registers 34 through 38 are internal to the MPI.

I/O Space

The MPI component enables an I/O board to act as a replier in I/O space (as seen in Figure 2). Board address space is programmed through interconnect space which allows multiple MPI-based boards to be used in a MULTIBUS II system with no jumpers. The width of the local I/O data bus can be 8 or 16 bits. Addresses and data for the local I/O is provided on a multiplexed bus.

Message Space

The MPI component enables an I/O board to send and receive interrupt packets, either in standard or Broadcast mode, without data. (Data transfer is carried out in I/O space.) Up to eight local interrupts may generate an interrupt packet onto the PSB; the highest priority interrupt level is encoded into this interrupt packet. The MPI entirely controls the access arbitration procedure for the PSB bus and the interrupt packet transfer. The MPI can receive interrupt packets from the PSB and uses them to generate a local interrupt signal.

The MPI interface supports parity signals when required and is capable of processing all error signals present on the bus.

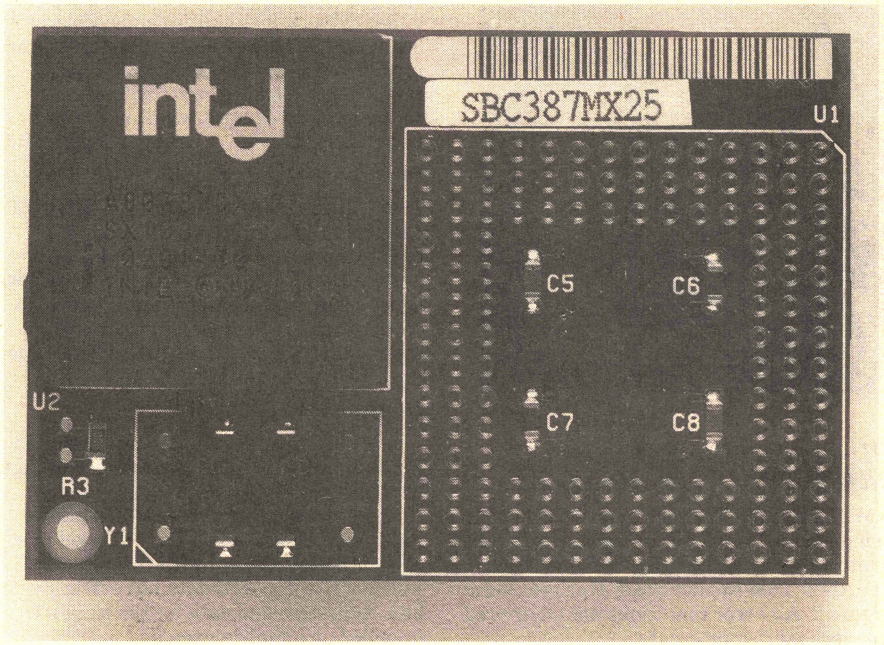
MPI SPECIFICATIONS

Power Supply Voltage: 0V–5V

Operating Temperature: 0°C–70°C

Storage Temperature: −65°C to +150°C

V_{CC} = 5.0V +10%



281009-52

iSBC 387MX25 MATH EXTENSION MODULE

- Adds i387™ Math Coprocessor floating point math capability to the iSBC 386/258 and the MIX 386/020A Multibus II boards
- Used in conjunction with existing i386™ microprocessor on the iSBC 386/258 or MIX 386/020A board. A separate i386™ Microprocessor is not supplied with the iSBC 387MX25 module
- Single slot solution—does not require an extra slot when the module is mounted on the board

SPECIFICATIONS

Clock Rates

i387™ Math Coprocessor	25 MHz
i386™ Microprocessor (not supplied)	16 or 20 MHz

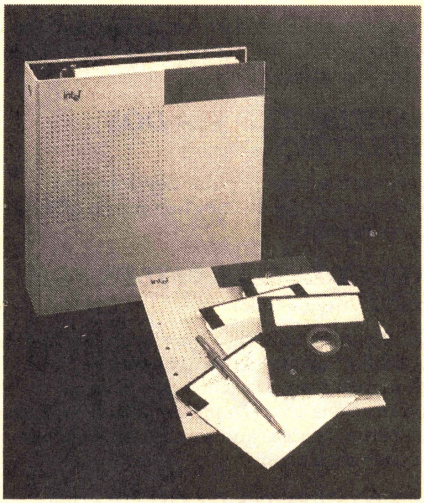
Physical Characteristics

Height:	1.8 inches
Depth:	2.7 inches
Width:	.43 inches

Power Requirements

Nominal Voltage (VDC)	Current (amps) Max	Power (watts) Max
+5	0.42	2.2

FIRMWARE DEVELOPMENT PACKAGE



281009-13

FIRMWARE DEVELOPMENT PACKAGE

The Multibus II Systems Architecture (MSA) provides the framework and guidelines for the start-up and run-time operation of a multiple board, multiple operating system, heterogeneous microprocessor, Multibus II system. The MSA Firmware Development Package (FDP) enables customers who are developing i386™ or i486™ CPU Multibus II boards to easily incorporate firmware on the board that allows it to fully participate in an MSA environment.

The FDP product includes source code in the C language for the generic initialization, diagnostic, and bootstrap elements of the Multibus II Systems Architecture. FDP provides a base level of code on which to begin development of board-specific MSA firmware and the C language source code makes the task of porting the FDP code to other CPU architectures less difficult.

Developing MSA firmware via FDP reduces implementation time and makes the benefits of MSA available at a fraction of the cost of developing a proprietary firmware solution.

FIRMWARE DEVELOPMENT PACKAGE FEATURES

- Source Code and Binary files for: Master Test Handler, Console Controller, Bootstrap Loader, Initialization and Diagnostic Executive, and Core Function Set
- Complete Documentation: Installation and Generation Guide, Functional Specifications, BIST Writer's Guide, MSA Specifications
- DOS generation environment
- Provided in C language
- Distributed via DOS diskettes, including make files compatible with PolyMake*
- Designed to facilitate customization

*PolyMake is a trademark of SAGE Software.

intel

FIRMWARE DEVELOPMENT PACKAGE FEATURES

INSTALLATION AND GENERATION GUIDE

The FDP Installation and Generation Guide provides an overview of MSA firmware and the FDP product. It also specifies the recommended tools and development environment for FDP.

SOURCE FILES

FDP is a source product. All files necessary to duplicate the MSA firmware functions for diagnostics, initialization, and booting are included in the package. Listings are also included for reference.

BINARY FILES

Binary files of each module are included and may make the generation of unmodified modules unnecessary. These object modules also serve as references in validating the development environment.

EXAMPLE BUILT IN SELF TEST (BIST) CODE

Examples of BIST code commonly used on Intel Multibus II hardware (e.g., RAM tests and message passing tests) are included for reference.

BIST WRITER'S GUIDE

A BIST Writer's Guide is included to aid the process of learning the BIST interfaces and to show the typical organization of BIST code on Intel hardware. Using the guide, the first time BIST writer will quickly come up to speed. Master Test Handler (MTH), Local Test Handler (LTH), and power-up Test Handler (PTH) interfaces to the Initialization and Diagnostics Executive (IDX) are covered.

SPECIFICATIONS PACKAGE

Detailed functional specifications for each FDP module are included. These documents are suitable for implementation purposes and were actually used in the development of base FDP firmware.

AVAILABLE IN C LANGUAGE

Recognizing the need for portability and the popularity of C, the FDP software is provided in the C language.

DISTRIBUTED VIA DOS DISKETTES

Covering the most popular development environment, the distribution media affords easy portability.

INCLUDES MAKE FILES FOR GENERATION

FDP includes a file that may be used with the PolyMake* utility to greatly ease the regeneration process. Only files having modifications are recompiled, which eases the generation process and significantly reduces the time required.

DESIGNED TO FACILITATE CUSTOMIZATION

The FDP code is written for Multibus II boards based on the 386 or i486 microprocessors. However, FDP is an open product which is partitioned and organized to facilitate changes and extensions necessary to support your hardware. In addition, since the source code is provided in the C language, the task of porting the FDP code to other microprocessors is less difficult.

TARGET HARDWARE REQUIREMENTS FOR DEVELOPMENT

Any open system that supports MSA will provide an adequate hardware environment for firmware validation.

PACKAGE CONTENTS

- FDP software on 5¼ inch DOS diskettes
- FDP Installation and Generation Guide
- BIST Writer's Guide
- FDP module functional specification documents
- MSA specification documents
- FDP-generated sample EPROM devices for the Intel iSBC 386/120 CPU board

*PolyMake is a trademark of SAGE Software

PRODUCT GUIDE

Product	Description
MIX BASEBOARDS	
MIX486DX66	MIXbaseboard with Intel486 DX2 CPU operating at 66 MHz
MIX486DX33	MIXbaseboard with Intel486 DX CPU operating at 33 MHz
MIX486SX33	MIXbaseboard with Intel486 SX CPU operating at 33 MHz
MIX EXPANSION MODULES	
MIX450	MIXTerminal Controller Module
MIX420	MIXWAN Module
MIX560	MIXEthernet Module
MIXSC10	Ten MIXStacking Connectors
MIX DEVELOPMENT MODULES	
MIXMOD1	Test Module
MIXMOD2	Breadboard Module
MIXMOD3	Debug Module
TERMINAL CONTROLLERS	
SBC186450	Mid-range terminal controller
MIX450 & MIXbaseboard	MIXTerminal Controller
WIDE AREA NETWORK CONTROLLERS	
SBC186410	Mid-range WAN board
MIX420 & MIXbaseboard	MIXWAN board
LOCAL AREA NETWORK/ETHERNET CONTROLLERS	
SBC186530	Mid-range Ethernet board
MIX560 & MIX baseboard	MIXEthernet Board with 1 MB
PERIPHERAL CONTROLLERS	
SBC386258SM01	Single-ended SCSI Controller w 1 MB
SBC386258SM04	Single-ended SCSI Controller w 4 MB
SBC386258DM04	Differential SCSI Controller w 4 MB
PARALLEL I/O	
SBCMPI519	Digital I/O Board
MULTIBUS II SILICON PRODUCTS	
82389	Message Passing Coprocessor
MPI	Multibus II Peripheral Interface
FIRMWARE DEVELOPMENT	
MSABASEFDP	Firmware Development Package
NETWORKING SOFTWARE	
INA960J	Networking Software

intel

PRODUCT GUIDE

WORLD WIDE SERVICE AND SUPPORT

Should your MIX or any Intel board ever need service, Intel maintains a world wide network of service and repair facilities to keep you and your customers up and running. For unique applications requiring customization of our products, the Intel Systems Group is available to modify, integrate and test Intel boards and system components to your requirements.

INTEL QUALITY—YOUR GUARANTEE

All MULTIBUS II I/O products are designed and manufactured to meet Intel's high quality standards. Intel quality is then verified by rigorous testing in our state-of-the-art Environmental Test Laboratory.

ORDERING INFORMATION

For more information or the number of your nearest Intel sales office, call 800-548-4725 (good in the U.S. and Canada).

UNITED STATES, Intel Corporation
3065 Bowers Ave., Santa Clara, CA 95051
Tel: (408) 765-8080

JAPAN, Intel Japan K.K.
5-6 Tokodai, Tsukuba-shi,
Ibaraki, 300-26
Tel: 029747-8511

FRANCE, Intel Corporation S.A.R.L.
1, Rue Edison, BP 303
78054 Saint-Quentin-en-Yvelines Cedex
Tel: (33) 1-30-57 70 00

UNITED KINGDOM, Intel Corporation (U.U>) Ltd.
Pipers Way, Swindon, Wiltshire, England SN3 1RJ
Tel: (44) (0793) 696000

WEST GERMANY, Intel Semiconductor GmbH
Dornacher Strasse 1, 8016 Feldkirchern bei Muenchen
Tel: (49) 089/90992-0

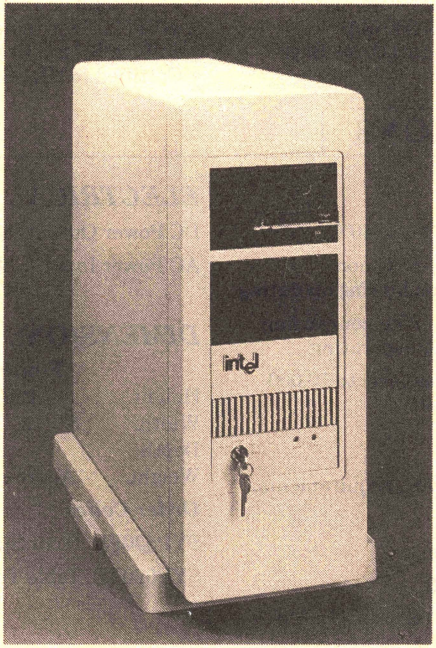
HONG KONG, Intel Semiconductor Ltd.
10/F East Tower, Bond Center,
Queensway, Central
Tel: (5) 8444-555

CANADA, Intel Semiconductor of Canada, Ltd.
190 Attwell Drive, Suite 500
Rexdale, Ontario M9W 6H8
Tel: (416) 675-2105

Multibus II Development Systems and Accessories

4

SYSTEM 520 CHASSIS FOR MULTIBUS II



280672-3

4

RUGGED AND DURABLE MULTIBUS II CHASSIS WITH EXCEPTIONAL EMI AND RFI SHIELDING

The eight slot SYP520FE and the SYP520TE provide a rugged and capable Multibus II chassis for integrated systems. Superior Intel quality is evident in the safety, EMI protection, and durability of the 520 chassis and integrated power supply. The eight slot Multibus II card capacity makes these chassis ideal for a wide variety of system applications.

The SYP520FE's and SYP520TE's capable features have made them the choice of many OEMs for development, test, and production systems. The SYP520FE and SYP520TE chassis deliver consistent performance in mission critical applications around the world. The chassis layout allows for good cooling, drive capacity, and serviceability.

*386 is a trademark of Intel Corporation.

October 1993
Order Number: 280672-004

intel

FEATURES

- Eight Slot Multibus II (IEE/ANSI 1296) Backplane
- 535 Watt Multiple Output Switching Power Supply Selectable for 115/230 VAC
- 3 Full Height (6 Half Height) Drive Bays with Front Panel Access
- Heavy Duty bus bars for negligible IR drop across card cage under maximum loading
- Floorstand or Tabletop models available (SYP520TE is table top unit and the SYP520FE is a floor stand model)
- Certified to UL CSA, IEC, and VDE

SPECIFICATIONS

ENVIRONMENT

Operating Temperature	10°C to 40°C
Relative Humidity	8–85% non condensing operating
	5–95% condensing non-operating
Altitude	Sea level to 10,000 feet

REGULATIONS

Meets or exceeds the following requirements:

Safety

US	UL 478 5th Edition
Canada	CSA C22.2 No. 220
Europe	IEC 380 and VDE 0806
EMI/RFIS & CANADA	FCC47 CFR Part 15 Subpart J Class B
Europe	VDE 0871 Level B

ELECTRICAL

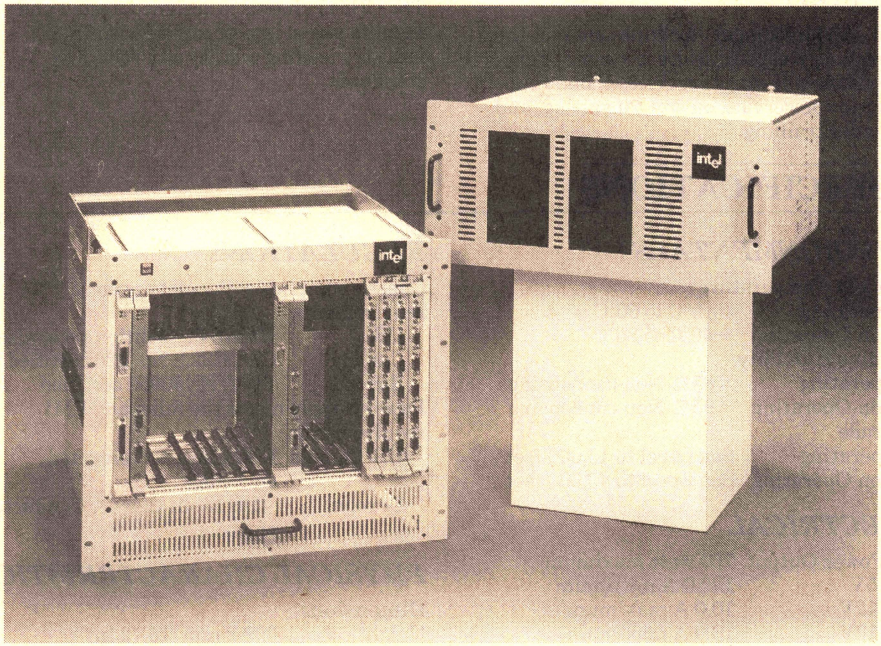
DC Power Output	535W Maximum
AC Power Input	88-132 VAC or 176-264vac; 47 Hz–63 Hz, single phase

DIMENSIONS

	Floorstand	Table Top
Height:	22.25″	8″
Width:	8″ (12″ at base)	17.5″
Depth:	23″	22.25″
Weight:	Approx. 70 lbs.	Approx. 60 lbs.

Order Codes

SYP520FE	Floor Stand Model
SYP520TE	Table Top Model



281029-1

A 20-SLOT MULTIBUS II CHASSIS FOR FLEXIBLE, EXPANDABLE SYSTEMS

Intel's 20-slot 541 CHASSIS provides a basic platform for the integration of large capacity systems. The CH541's modular packaging allows for integration into standard 19-inch rack-mount cabinets and fits in a 24-inch deep rack. Each CH541 chassis is compactly built, ruggedly constructed, and features superior Intel quality and high reliability.

FEATURES

- 20-slot Multibus II (IEEE/ANSI 1296) backplane; EIA, 19-inch rack standard with provision for slide mounting/fits 24-inch deep rack
- Heavy duty bus bars for negligible IR drop across card cage under maximum loading
- 750 Watt multiple output switching power supply with 120A at 5 VDC; jumper selectable for 115/230 VAC
- CH541 Chassis certified as a complete unit to UL, CSA, and TUV safety agency requirements
- High-volume and uniform cooling air flow across all card positions (250 + LFM). Air flow highly independent of adjacent equipment
- P2 backplane at Slot 0 provides "Reset," Remote Diagnostics Serial Port Interface (RSPI) functions and I/O connections
- Unrestricted area behind P2 backplane for user I/O configuration
- One Year Full Warranty

September 1991
Order Number: 281029-002

intel

FEATURES

WORLDWIDE SERVICE AND SUPPORT

Intel provides support for Intel and non-Intel boards and peripherals. Development support options include on-site service, phone support, subscription service, on-site consulting, and customer training.

QUALITY AND RELIABILIITY

The CH541 chassis is designed, tested and manufactured in accordance with Intel's industry leading quality and reliabilty standards.

SPECIFICATIONS

ENVIRONMENTALS

Ambient Temperature	
Operating	0°C to 50°C
Non-Operating	−40°C to 85°C
Relative Humidity	
Operating	<85% Non-condensing
Non-Operating	<95% Non-condensing
Altitude	
Operating	Sea Level to 10,000 feet
Non-Operating	Sea Level to 40,000 feet

ELECTRICAL

DC Power Output	750 watt maximum
+5V	120.0 A maximum
+12V	10.0 A maximum
−12V	10.0 A maximum
AC Power Input	90-132 VAC at 12 A or 180-264 VAC at 6 A 47-63 Hz

REGULATIONS

Chassis complies with safety agency requirements:

US	UL478 5th Edition recognized
Canada	CSA C22.2 No. 220 certified
Europe	TUV, EN60950 certified

Power Supply meets the following EMI requirements:

US/Canada	FCC Class A Conducted emissions
Europe	VDE Limit Class A Conducted emissions

PHYSICAL CHARACTERISTICS

Dimensions

Height	399.2 mm (15.72 in)
Width	482.7 mm (19.00 in)
Depth	497.8 mm (19.60 in)
Weight	15.9 kilograms (35 lbs)

ORDER CODE

CH541C20R

SYSTEM ENCLOSURES AND BACKPLANES

A LOW COST RACKMOUNT PERIPHERAL CHASSIS FOR FLEXIBLE, EXPANDABLE SYSTEMS CONFIGURATIONS

Intel's 542 Rackmount Peripheral Chassis provides a basic platform for integrating large capacity systems. The CH542 chassis is a companion to Intel's CH541 20-slot chassis. However the CH542 chassis may be used with any system where additional peripheral capacity is required. Intel's modular packaging allows for integration into standard 19-inch rackmount cabinets and fits in a 24-inch deep rack. Built into each CH542 chassis is rugged compact construction, superior Intel quality, and high reliability.

FEATURES

- Rigid light weight structure to fit EIA 19-inch rack standard with provision for slide mounting, fits 24-inch deep rack.
- Three full height bays (2 front access, 1 internal) each able to mount two half height devices.
- 300 Watt multiple output switching power supply with 10A at + 12 VDC. Switch selectable for 115/220 VAC.
- CH542 Chassis designed to comply with international safety agency requirements.
- Air flow capable of supporting 40°C operation.
- Power distribution cables provided for all drive positions plus an adaptor board.
- Cable manager with strain relief for rear entry signal cables.
- One year full warranty.

intel

FEATURES

ONE YEAR WARRANTY

All Intel board level products have a one year mail-in warranty on both parts and labor.

WORLDWIDE SERVICE AND SUPPORT

Should any Intel enclosure or backplane ever need service beyond the warranty period, Intel maintains a worldwide netork of service and repair facilities to keep you and your customers up and running.

QUALITY AND RELIABILIITY

The CH541 and CH542 are designed, tested and manufactured in accordance with Intel's industry leading quality and reliabilty standards.

SPECIFICATIONS

ENVIRONMENTALS

Ambient Temperature	
Operating	0°C to 40°C
Non-Operating	−40°C to 85°C
Relative Humidity	
Operating	<85% Non-condensing
Non-Operating	<95% Non-condensing
Altitude	
Operating	Sea Level to 10,000 feet
Non-Operating	Sea Level to 40,000 feet

ELECTRICAL

DC Power Output	300 watt maximum
+5V	26.0 A maximum
+12V	10.0 A maximum
AC Power Input	90-132 VAC at 5 A or 180-264 VAC at 2.5 A, 47-63 Hz

REGULATIONS

Chassis designed to comply with international safety agency requirements.

Power Supply meets the following EMI requirements:

US/Canada	FCC Class A Conducted emissions
Europe	VDE Limit Class A Conducted emissions

PHYSICAL CHARACTERISTICS

Dimensions	
Height	22.1 cm (8.72 in)
Width	48.3 cm (19.00 in)
Depth	48.0 cm (18.91 in)
Weight	9.1 kilograms (20 lbs)

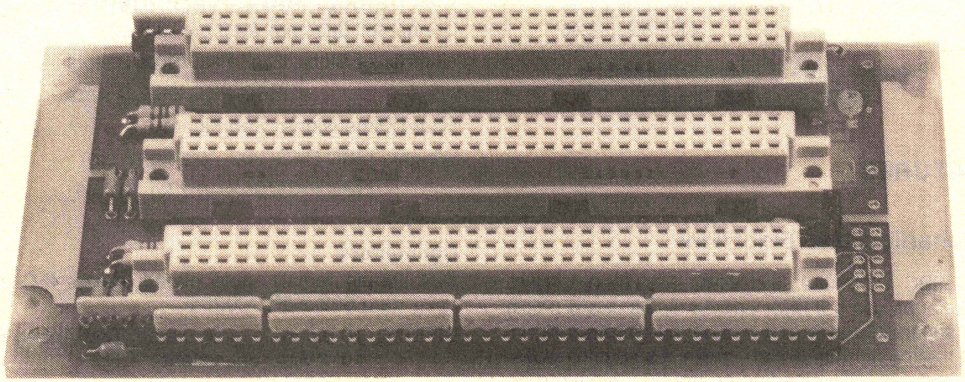
ORDER CODE

CH542R

iSBC PKG/902
iSBC PKG/903
MULTIBUS II iLBX II BACKPLANES

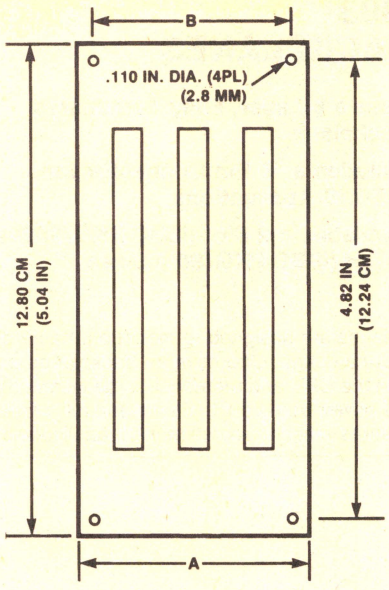
- **Provides iLBX II Interconnect for Fastest CPU/Memory Data Transfers**
- **Designed to Mount in Multibus II Cardcage Assemblies**
- **Meets All Electrical and Mechanical Requirements of the Multibus II Specifications**
- **Uses a 6 Layer, Fully Terminated Backplane**
- **Includes a 10 Pin Connector for BITBUS Applications**
- **Available in 2 Slot (iSBC PKG/902) and 3 Slot (iSBC PKG/903) Sizes**

The iSBC PKG/902 and iSBC PKG/903 series of iLBX II backplanes are designed to mount on the P2 side of Intel's Multibus II cardcage assembly or other double Euro (6U) cardcage. One or more backplanes may be installed in a system to allow high speed data transfers between the CPU and memory boards installed in the system. The iLBX II backplane uses a 6 layer PCB with separate power and ground planes and full termination on all signal lines. This design minimizes system noise and ensures reliable operation in all applications.



280074-1

August 1993
Order Number: 280074-002



Dimensions		A	B
iSBC PKG/902	IN	1.55	.80
	CM	3.94	2.03
iSBC PKG/903	IN	2.35	1.60
	CM	5.97	4.06

Figure 1. iLBX II Board Dimensions (iSBC PKG/903 Shown)

FEATURES

Mechanical and Electrical

The iSBC PKG/902 and iSBC PKG/903 iLBX II backplanes use a 6 layer printed circuit board (PCB) with separate power and ground layers and a signal lead routing scheme which minimizes ringing, crosstalk, and capacitive loading on the bus. Mounted on the PCB are two (iSBC PKG/902) or three (iSBC PKG/903) 96 pin DIN connectors, one 10-pin BITBUS connector, terminating resistors, decoupling capacitors, and power terminals. The resistors and capacitors are mounted into sockets, and all parts are press-fit into the backplane. The PCB is UL recognized for flammability.

Single In-line Package (SIP) style resistors are used to terminate all address, clock, data, and control lines. Each termination consists of two resistors which connects the line to +VCC and ground. Different size resistors are used according to the type of driver connected to the line in an operating system. The SIP style resistors help make the board compact in size and allows the designer to mount several backplanes directly adjacent to one another in a system without having to skip slots.

Mounted on the rear of the backplane is a 10-pin BITBUS connector. This connector serves as the serial communication interface for any iSBX 344 BITBUS controller boards installed in the system.

The DIN type connectors are female, 96 pins, fully gold plated, and meet IEC standard 603-2-IEC-C096F. The connectors are mounted on 0.8″ centers to match Intel's iPSB (Parallel System Bus) Multibus II backplanes and are keyed to ensure proper mating to the Multibus II board. The connector can provide up to 6 amps of current at +5V to each Multibus II board in addition to the current available over the Parallel System Bus backplane.

Screw terminals on the backplane are provided for connection to +5V power and ground. These terminals, each of which can handle up to 25 amps of current, provide a simple and highly reliable connection method to the power supply.

SPECIFICATIONS

Mechanical and Environmental

Connector Spacing: 20.3 cm (0.8 in)

Number of Slots: iSBC PKG/902: 2 slots
iSBC PKG/903: 3 slots

Board Dimensions: See Figure 1

Weight: iSBC PKG/902—0.2 kg (8 oz)
iSBC PKG/903—0.3 kg (12 oz)

Connectors:

DIN: 96-pin female, gold plated, meets IEC standard 603-2-IEC-C096-F

BITBUS: 10-pin male, gold plated, T&B Ansley 609-1012M, or equivalent

Construction Method: Six layer backplane with separate VCC and ground layers

All connectors, power terminals, and resistor/capacitor sockets are press-fit into the backplane

Mounting Hole Location: See Figure 1

Operating Environment: 0°C–70°C ambient temperature; 0% to 90% relative humidity, non-condensing; 0 ft.–10,000 ft. altitude

Electrical

Backplane Electrical Characteristics and Line Terminations:	Per Intel Multibus II specification 146077, Sec. II, iLBX II

Power Connections

Type: Screw terminal block: AMP P/N 55181-1; Winchester P/N 121-25698-2; or equivalent

Mating Connection: No. 6 locking spade or ring tongue lug

Quantity: 2(VCC, Ground)

Current Rating: iSBC PKG/902: 12 amps; iSBC PKG/903: 18 amps (Power and Ground)

Maximum Current Available Per Slot: 6 amps (over the iLBX II backplane)

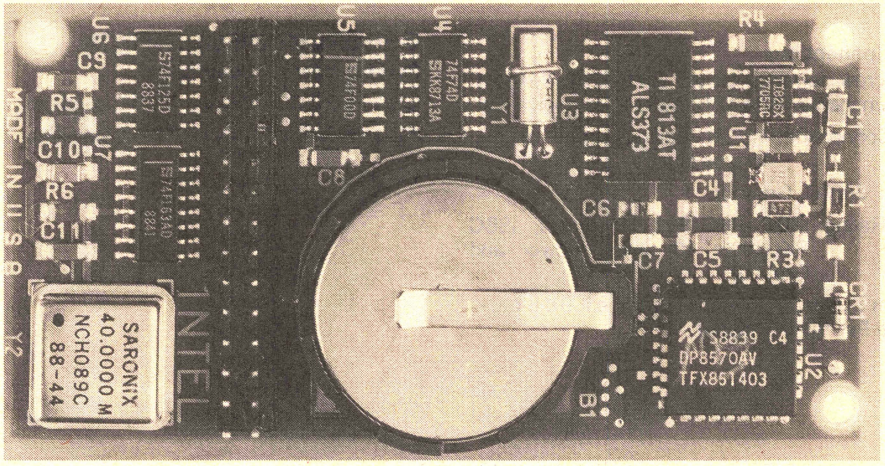
REFERENCE MANUAL

Multibus II Cardcage Assembly and iLBX Backplane User's Guide, P/N 146709-001 (not supplied)

ORDERING INFORMATION

Part Number	Description
iSBC PKG/902	2 slot iLBX II Backplane
iSBC PKG/903	3 slot iLBX II Backplane

iSBC CSM/002 CENTRAL SERVICES MODULE



280694-1

COST REDUCED CENTRAL SERVICES MODULE

The iSBC CSM/002 module is a small, surface mount circuit board which performs all central service module (CSM) functions as required by the IEEE/ANSI 1296 Multibus II specification. This credit card sized module mounts on a compatible base board such as the iSBC 386/258, iSBC 386/133, iSBC 486/125, or iSBC 486/133SE single board computer. The combined host board and CSM module require only one card slot. The small size and high functionality of the iSBC CSM/002 module is achieved by taking advantage of silicon support for CSM functions on the MPC (Message Passing Coprocessor) bus interface component. This module reduces system cost while remaining software compatible with the previous iSBC CSM/001 board.

FEATURES

- Full IEEE/ANSI 1296 Compliance for CSM Functions:
  - — Arbitration and slot ID Initialization
  - — BCLK and CCLK Generation
  - — PSB Bus Timeout Monitoring
  - — Reset Sequencing for Warm and Cold Resets
  - — Power Fail Indication and Recovery
- Software compatible with the iSBC CSM/001 board, but saves a card slot
- Battery Backup Time-of-day Clock
- Slot 0 Detection Circuit
- Clock Based Alarm Function for Periodic Interrupt
- 28 Bytes Non-volatile RAM
- Chassis ID for Crate-to-Crate Addressing
- Low Battery and Oscillator Failed Warnings

August 1993
Order Number: 280694-002

FEATURES

WHAT IS A CENTRAL SERVICES MODULE?

The Central Services Module centralized a variety of bus management tasks in a Multibus II based system, reducing system overhead:

- Sequencing of reset signals on the PSB backplane.
- Assignment of card slot and arbitration IDs.
- Supplies a 10 MHz, system-wide clock signal (BCLK).
- Monitors the PSB bus for time outs, and signals a bus error when a parity error is detected.

FULL IEEE 1296 COMPLIANCE

The iSBC CSM/002 module meets all timing requirements for Central Services Module functions according to the Multibus II specification. This ensures reliable, clean system clock signals and correct reset sequencing for system power-on, power-fail, and front panel warm and cold resets. When used in a system where the power supply is capable of generating an ACLO indicator, the CSM module will signal a non-maskable interrupt to the host CPU shortly before the power goes down.

COMPACT SIZE: NO LONGER REQUIRES A SEPARATE CARD SLOT

The iSBC CSM/002 module reduces total system cost by supporting all CSM functions in an inexpensive, credit card sized module. This module mounts component side down onto compatible baseboards like the iSBC 386/258, iSBC 386/133, iSBC 486/125, or iSBC 486/133SE single board computers which have built-in CSM connectors. The combined host board and CSM module occupy only one card slot. The iSBC CSM/002 module is fully software compatible with the earlier iSBC CSM/001 board, and is a direct replacement. Only one CSM module is required per system.

TIME OF DAY CLOCK FUNCTIONS

A battery backed up time-of-day clock is supplied on the iSBC CSM/002 module. This feature is software compatible with existing Time and Date commands supported by various Intel supplied operating systems. In addition a new periodic alarm function is now available. This feature allows the user to generate an interrupt to the local processor based on the system clock. Intervals can be selected ranging from one second to one year. One example of how to use this might be to schedule a disk backup to tape at 1:00 am on Friday of each week.

NON-VOLATILE RAM FUNCTIONS

A two byte chassis ID is stored in an interconnect register in order to identify a particular backplane segment in a network which consists of a large number of nodes distributed in multiple chassis. In addition, there are 28 bytes of user definable non-volatile RAM available. One application might be for a bootstrap password to prevent unauthorized access to a system.

PROGRAMMATIC INTERFACE

All access to the above functions is via a set of function records contained in interconnect address space. These registers are resident on the host in slot space. These registers are resident on the host in slot zero, but are accessible to any agent on the PSB. This allows other boards to query reset status, bus errors, system time, NVRAM contents, and many other centralized functions.

WORLDWIDE SUPPORT AND SERVICE

Assistance in developing and supporting Multibus II applications is available through Intel's network of field application engineers, system engineers, customer training centers and service centers.

INTEL QUALITY—YOUR GUARANTEE

The iSBC CSM/002 module is designed and manufactured in accordance with Intel's high quality standards. Quality is verified by rigorous testing in Intel's state-of-the-art Environmental Test Laboratory.

intel

ORDERING INFORMATION

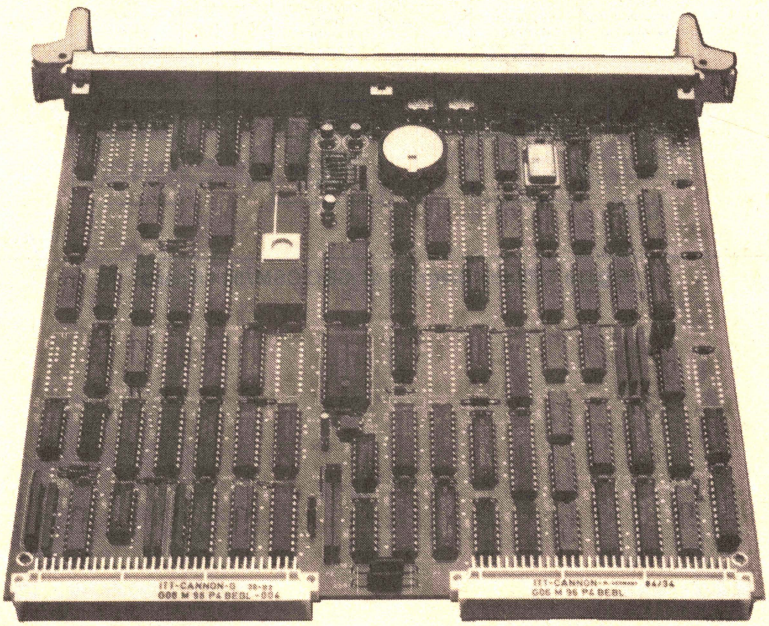
Code	Description
SBCCSM002	iSBC CSM/002 Hardware Reference Manual P/N 459706-001

AP NOTE: "Design of a Cost Reduced Central Services Module for Multibus II"

iSBC CSM/001
CENTRAL SERVICES MODULE

- **iSBC CSM/001 Central Services Module Integrates Multibus II Central System Functions on a Single Board**
- **Multibus II Parallel System Bus Clock Generation for all Agents Interfaced to the Multibus II PSB Bus**
- **System-wide Reset Signals for Power-up, Warm Start, and Power Failure/ Recovery**
- **System-wide Time-out Detection and Error Generation**
- **Slot I.D. and Arbitration I.D. Initialization**
- **Multibus II Interconnect Space for Software Configurability and Diagnostics**
- **Built-In Self Test (BIST) Power-up Diagnostics with LED Indicator and Error Reporting Accessible to Software via Interconnect Space**
- **Time-of-day Clock Support with Battery Back-up on Board**
- **Double-high Eurocard Standard Form Factor, Pin and Socket DIN Connectors**

The iSBC CMS/001 Central Services Module is responsible for managing the central system functions of clock generation, power-down and reset, time-out, and assignment of I.D.s defined by the Multibus II specification. The integration of these central functions in a single module improves overall board area utilization in a multi-board system since these functions do not need to be duplicated on every board. The iSBC CSM/001 module additionally provides a time-of-day clock.



280070-1

August 1993
Order Number: 280070-004

FUNCTIONAL DESCRIPTION

Overall

The iSBC CSM/001 Central Services Module integrates Multibus II central system functions on a single board. Each Multibus II system requires management of these central system functions as defined in the Multibus II specification. To perform its central system functions, the iSBC CSM/001 Central Services Module has a fixed slot I.D. and location in the backplane. The iSBC CSM/001 board additionally provides a time-of-day clock.

Architecture

The iSBC CSM/001 board is functionally partitioned into 6 major subsystems. The Central System Wide Control subsystem includes Multibus II PSB bus clock generation and system wide reset signal generation. The Time-Out Control subsystem provides system wide time out detection and error generation. The System Interconnect Space subsystem controls I.D. initialization and software configurable interconnect space. The Link Board interface subsystem provides links to other buses. The last two subsystems are of the Time-of-Day clock and the PSB bus interface. These areas are illustrated in Figure 1.

CENTRALIZED SYSTEM-WIDE CONTROL SUBSYSTEM

Parallel System Bus Clock Generation

The CSM generates the Parallel System Bus clocks. The Bus Clock (BCLK*) 10 MHz signal and the Constant Clock (CCLK*) 20 MHz signal are supplied by CSM to all boards interfaced to the Parallel System Bus. These boards use the Bus Clock 10 MHz signal for synchronization, system timing, and arbitration functions. The Constant Clock is an auxiliary clock. The frequency of the Bus Clock and Constant Clock can be halved via jumpers for diagnostic purposes.

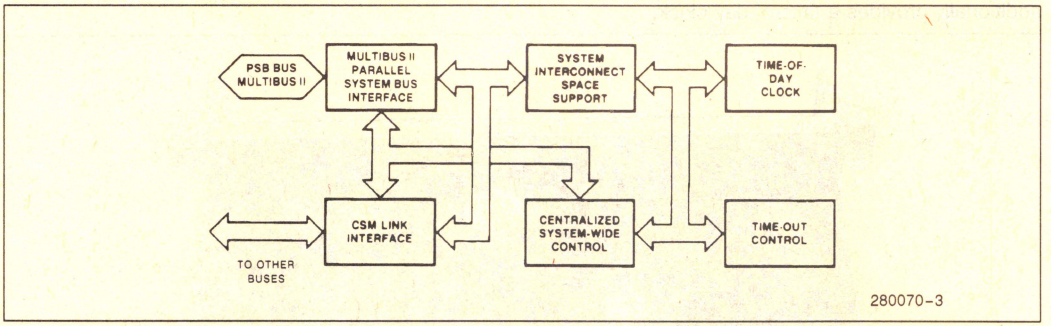


Figure 1. Block Diagram of iSBC CSM/001 Board

TIME-OF-DAY CLOCK SUBSYSTEM

The Time-Of-Day Clock subsystem consists of a clock chip, battery, and interface circuitry. The clock provides time keeping to 0.01% accuracy of fractions of seconds, seconds, minutes, hours, day, day of week, month, and year. This information is accessible via the interconnect space. The battery back-up for the clock chip provides 2 years of operation.

SPECIFICATIONS

System Clocks

BCLK* (Bus Clock)	10 MHz
CCLK* (Constant Clock)	20 MHz
LCLK* (Link Clock)	10 MHz

Jumper option available to divide these frequencies in half

Interface Specifications

Location	Function	Part #
P1	PSB Bus	603-2-IEC-C096F
P2	Link and Remote Services	603-2-IEC-C064-F

PHYSICAL DIMENSIONS

The iSBC CSM/001 board meets all Multibus II mechanical specifications as presented in the Multibus II specification.

Double-High Eurocard Form Factor:

Depth:	22.0 cm. (8.7 in.)
Height:	23.3 cm. (9.2 in.)
Front Panel Width:	2.0 cm. (0.78 in.)
Weight:	0.5 kg. (16.5 oz.)

ENVIRONMENTAL REQUIREMENTS

Temperature:	(inlet air) at 200 LFM airflow over boards	
	Non-operating:	−40 to +70°C
	Operating:	0 to +55°C
Humidity:	Non-operating:	95% RH @ 55°C
	Operating:	90% RH @ 55°C

POWER REQUIREMENTS

Voltage (volts)	Current (amps)
+5 V_{CC}	6A (max.)
+5 (battery)	0.65A (max.)

BATTERY CHARACTERISTICS

3V nominal voltage; capacity of 160 milliamp hours minimum.

BATTERY DIMENSIONS

Outside dimension	20 mm–23 mm
Height	1.6 mm–3.2 mm

REFERENCE MANUAL

iSBC CSM/001 Board Manual (# 146706-001)

ORDERING INFORMATION

Part Number	Description
SBC CSM/001	Multibus II Central Services Module

Multibus II Architecture

5

The Multibus II Bus Structure

INTRODUCTION

Many people equate the phrase "Multibus II" with the Parallel System Bus defined within the IEEE/ANSI 1296 specification. While this over simplification is often useful, the failure to appreciate that *it is a contraction* of a more embracing architecture can lead one astray when comparing the Multibus II bus structure with other buses. Comparisons between the Multibus II Parallel System Bus and other buses are often completed in isolation, without full regard to the framework in which the Multibus II architecture was defined. This chapter rebuilds this framework, describes its hierarchical structure and details how its features are *required* for multiple microprocessor designs of today.

CUSTOMER NEEDS DEFINE THE NEW BUS STRUCTURE

Intel Corporation had had many years experience with the Multibus standard before embarking upon the requirements for a "next generation" bus structure. The first Multibus standard bus was introduced in 1974 and it was fundamentally a CPU/Memory bus. It evolved along with microprocessor technology to become a multi-master shared memory bus capable of solving most real time applications of the 1980s. The silicon trends throughout the 1980s were dramatic with DRAM densities increasing by a factor of two every three years, so projecting exactly what customers would require in the late 1980s and through to the early 2000s was particularly difficult. Intel therefore set up a consortium with eighteen of its larger customers and other industry leaders who could see the potential within the single board computer industry, to define the scope and possibilities of what was to be call "Multibus II".

It was known that the rate of silicon integration would allow a complete computer system including CPU, Program Memory, Data Memory, Input/Output and bus interface to be fabricated upon a single board. With such a large transistor budget to be spent upon implementing a single board computer, where would be the optimal places to best utilize the technology? Self test and diagnostics could now be considered—with so much silicon on a board it would be prudent to use some of the transistor count to TEST the remainder of the board. Since board manufacturers are integrating more and more VLSI silicon onto their boards, the user needs some reassurance that the basic board functionality is intact before they load their value added code—the user is *demanding* on-board diagnostics for these highly integrated boards. The bus interface itself, not a traditional candidate for high integration silicon circuitry, could use transistors for added sophistication IF this sophistication could make the single board computers easier to use. A trend began to develop; transistors added to improve ease of use filtered to the top of the implementation list.

With the increased silicon densities available, semiconductor manufacturers turned their focus upon increased capability peripheral components. Their use on single board computers served to compound the boards complexity and the single board computer user was "rewarded" by having to wade through lengthy reference manuals and innumerable jumper options—often arriving at the final solution only by trial and error. Memory mapping options, arbitration priorities, interrupt levels and scores of other "tunable" parameters contributed to the hassle, leaving the systems engineer confused and amazed. Often the only solution was to locate a board which had already been properly configured and was operating and then copy off the jumper list.

Board manufacturers built in numerous options so that their products could be used in the broadest possible spectrum of applications. The number of options offered was not the core of the problem—*but managing them was.* Options allow interrupt routing, memory mapping, EPROM size selection, timing and other user installed components. When the jumper count exceeded 200, it no longer made sense to monopolize board real estate since an inexpensive microcontroller could be used to manage the resources more effectively.

A system bus requires standardized system-wide configuration information to be made accessible to software, opening it to opportunities for centralized control and coordination. Ideally the end-user of these products will be completely unaware of the configuration process. They simply remove the board from it's shipping container, plug it into any free slot in the backplane, and apply power. Things work the first time around with no mess, no fuss, and no configuration errors.

September 1993
Order Number: 281113-001

The consortium therefore placed focus on the system aspects of the single board computer design. The developing model for a typical system built from these highly capable single board computers was based upon functionally partitioned subsystems interacting across a standardized communications channel. This precipitated a change in philosophy for the traditional system development from the single board computer outward to a higher level systems perspective, specified topsdown and bound together by rigid interfaces.

The consortium quickly reached consensus that no single bus could be used to satisfy all aspects of a design of this type—too many variables would have to be compromised, so a *multiple bus* structure was defined in a similar fashion to its Multibus I (IEEE 796) predecessor. Figure 1 shows the four sub- buses defined by the consortium: the iSBX bus was retained for incremental I/O expansion, a local CPU/Memory expansion bus was proposed and two versions of a SYSTEM bus (serial and parallel) were defined. The concept of a SYSTEM bus is an important one to grasp—all open buses to date were basically CPU/Memory buses with little regard for system aspects. To have an open bus *SPECIFICALLY* designed to be a system bus was a bold step.

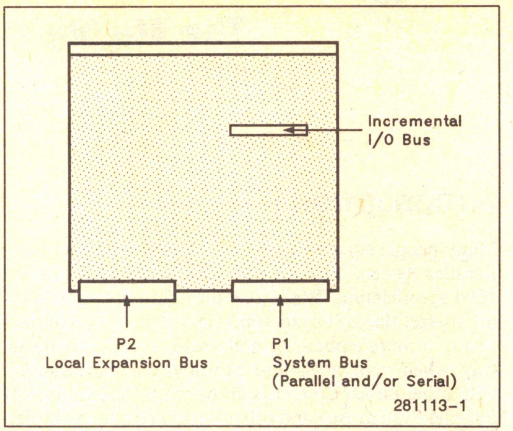


Figure 1. Since no single bus could solve the defined problem set, a multiple bus solution was proposed

FUNCTIONAL PARTITIONING AS A SOLUTION FOR NON-OBSOLESCENCE

Before detailing the attributes of each of the defined buses that make up the Multibus II systems architecture it is important to appreciate the model developed for the bus. Figure 2 shows a typical collection of systems connected to a local area network or LAN. This type of networked systems solution is very popular with

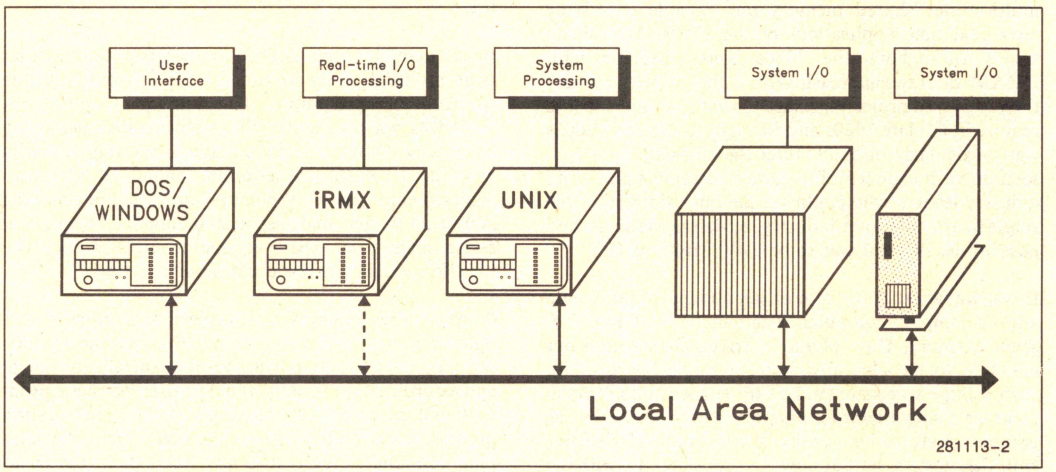


Figure 2. Multiple diverse systems can co-exist on a Local Area Network using defined protocols

systems builders since it boasts a large array of benefits. The solution is functionally partitioned—separate systems are used to tackle different facets of the overall problem. These systems are independent from each other and decisions made to optimize each of them for their individual task may be made in isolation with respect to the other systems in the network. This degree of freedom gives the systems architect an unquestioned edge when engineering tradeoffs are being made. The choice of hardware, options and software may be made with the sole goal of solving the small part of the overall problem currently in focus. Each system is typically tuned for its task using specifically configured hardware and software and it is not uncommon to see multiple different operating systems with a single network. Systems that *MUST* respond in real time, for example, would use Intel's iRMX® Real Time operating system or their iRMK Real Time kernel, while a data base manager would use an industry standard operating system such as UNIX or DOS/Windows. The systems would probably contain a diversity of microprocessors.

Each of the systems are individually upgradeable. If something bigger/better/faster/cheaper becomes available it is easy to integrate this into a networked system solution. New technology may be applied at strategic points on the network and no major overhaul of the complete solution is ever required. Since the overall system is continuously upgradeable it will not become obsolete and will serve for many years.

The systems are independent in their own right, capable of completing their assigned task in isolation and need not be connected to the LAN to function. The reason that they are connected to the LAN is to enable the sharing of data. The LAN defines a media type and details certain communications protocols that all the systems on the network must adhere to—in this way the diverse systems may share data in a consistent manner. Each system will require a hardware interface to the media and a software interface to the network protocols. Error recovery and retry algorithms are employed to ensure reliable communications between the individual systems on the network.

The software model for this functionally partitioned solution is "protocol based" with "data movement". In this type of model the computer population is split into server systems and consumer systems. A server system provides facilities and resources to the network such as file systems or access to a communications hierarchy. A consumer system does work using the facilities provided by the network servers. The consumer software model makes defined requests for data that a server will respond to. An inter-system communications standard was developed by the International Standards Organization and its seven-layer model is shown in Figure 3. All interfaces are rigidly defined but the implementation is not—this allows many diverse systems to interact successfully.

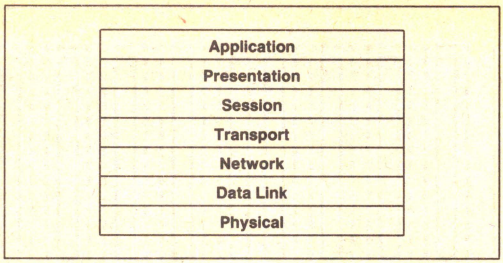


Figure 3. A seven layer inter-system communications model has been defined by the International Standards Organization

A MAJOR BREAKTHROUGH

Imagine now, keeping the same networked topology but using the advances in silicon integration to compress the systems into a single Multibus II chassis as shown in Figure 4. We will use the parallel system bus as the high speed, reliable network media. We will use protocols on the parallel system bus very similar to the protocols used over the local area network. Each of the networked systems will become a single board subsystem (or a multi-board subsystem if the circuitry exceeds the area of a single board). We have created a VERY Local Area Network or a "Network-in-a-Box".

A Multibus II chassis with multiple boards operating as a "Network-in-a-Box" is physically similar to a traditional system but its networked-subsystems philosophy realizes many benefits. This use of advanced Multibus II bus technology allows the systems builder to tackle multi-CPU designs with confidence.

Partitioning a multi-CPU application into a set of networked subsystems allows us the opportunity to focus. We are able to break down a complex problem into solvable sub-problems. These sub-problems are also encapsulated, therefore they can be solved independently of the other problems. Encapsulation is an important attribute of a Multibus II subsystem. Encapsulation assumes that a subsystem is intelligent and that it owns most of the resources required to complete its assigned task. These resources are private and are under the sole control of the subsystem CPU(s). In a complex subsystem these resources may be spread over multiple boards as shown in Figure 5. Even though there are three physical boards in this example it is important to appreciate that, at a systems architecture level, they are treated as a single logical subsystem. The boards within a subsystem can communicate using a local expansion bus defined on the P2 connector.

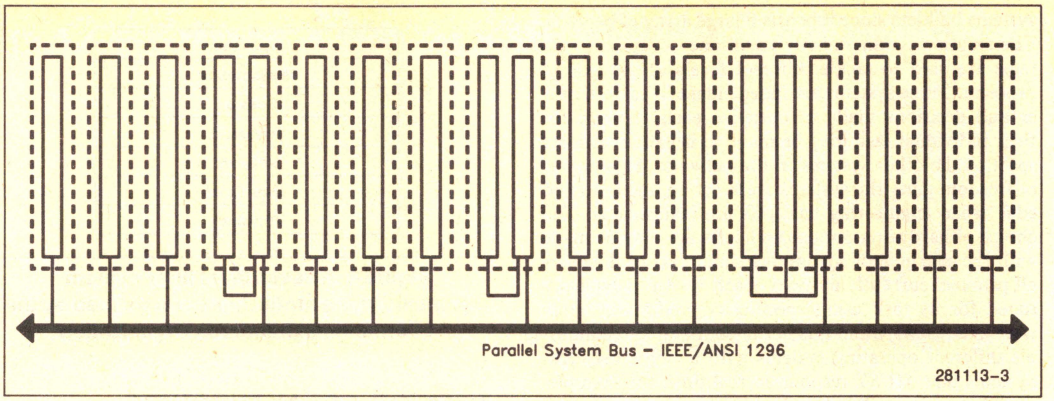


Figure 4. Higher levels of silicon integration allow a network topology to be used within a single system

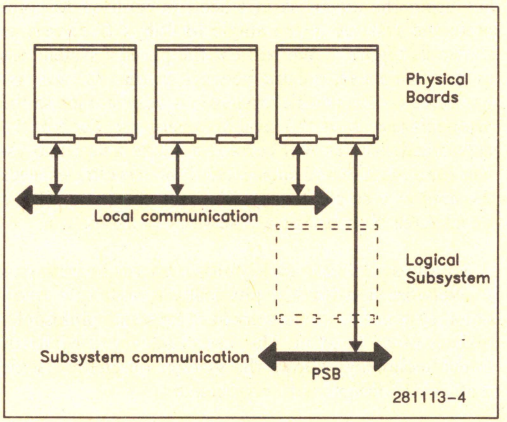


Figure 5. Multi-board subsystems interact with each other across a private P2 bus and appear as a single logical subsystem

IMPLEMENTING THE SUB-BUSES

The realization that a single bus could not solve all of the problems is an important first step. The requirements of each sub-bus are so different that compromising their features will result in sub-optimal system solution. The majority of this chapter will detail the system bus, but the other buses are discussed so that a context for decisions made will be evident.

The Incremental I/O bus needs to be simple. Its role is to allow the addition of a small piece of input/output onto a single board computer to customize it for a particular application. Performance is not an issue but low interfacing costs are. More extensive I/O would be added on the local expansion bus or on the system bus if an accompanying microprocessor was appropriate.

The Local CPU/Memory Expansion bus will always be dependent upon microprocessor technology. The interface between a CPU and its memory needs to be tightly coupled if we are to extract the maximum performance levels from a given microprocessor family. This bus will evolve with microprocessor technology and will typically exist for only two to four years before it has to be redesigned. If the CPU element requires more MIPs then additional identical microprocessors could be closely coupled on this local expansion bus; if these microprocessors had on-chip or local caches, as many of the higher performance offerings do, then this multiple microprocessor CPU/Memory bus must be cache coherent.

A major requirement of the SYSTEM bus is a technology independent communications media. Since this bus will remain constant throughout multiple generations of microprocessors it must be decoupled from the microprocessor technology used on the single board computer. This loosely coupled approach, whereby each single board computer subsystem is independent, will enjoy all of the benefits of the systems networked on a Local Area Network. Global system functions such as initialization, diagnostics and configuration must be added in a standardized way to this long-lived system bus.

PHYSICAL STANDARDS

A reasonably large card size with ample power is key to making the best use of the available levels of silicon integration. While no real data has proven that edge connectors should not be used, there is a definite trend towards gas-tight pin-and-socket connectors. A double Eurocard format, IEEE/ANSI 1101 Standard, with dual 96 pin DIN connectors was chosen for the Multibus II standard. A 'U' shaped front panel, licensed from Siemens, West Germany, was chosen for its enhanced EMI and RFI qualities.

THE INCREMENTAL I/O BUS

The large array of existing iSBX (IEEE 894) modules for the Multibus I family of products encouraged its adoption within the Multibus II standard. The iSBX strategy has proven itself with customers and vendors alike.

For more advanced I/O needs, the Modular Interface eXtension (MIX) Architecture was created. For more information on MIX, refer to Application Note, AP-423,

THE LOCAL EXPANSION BUS

The exact bus used for local expansion will vary according to the specific requirements and performance levels required in a subsystem design. As far as the IEEE/ANSI 1296 specification is concerned, this is an open option and ANY bus that is suitable may be used. Intel initiated a standard call iLBX II which was optimized for a 12 MHz 80286 microprocessor although other manufacturers have implemented this using members of the 68000 family. Siemens have implemented Multibus I on the P2 connector and called it the AMS bus. Intel has also offered the PC AT bus as a subsystem option on a range of PC compatible products—while this subsystem bus is low performance, it is a low cost method to add dumb I/O to a Multibus II subsystem. The IEEE 896 committee is currently working upon cache coherent extensions to Futurebus; this bus, discussed in Chapter 7, would be a good candidate for a high performance local expansion bus.

Address Space		Address Space Size	Sequence Type	Transfer Width (bits)	Block Transfers	Number of Repling Agents
CPU/Memory Space	Memory	2**32 bytes	Read/Write	8, 16, 24, 32	Supported with increment	One
	Input/Output	2**16 8-bit ports	Read/Write	8, 16, 24, 32	Supported without increment	One
System Space	Message	2**8-1 Agents 1 Broadcast	Write Only	32	Supported without increment	One or All
	Interconnect	2**9 8-bit registers each agent	Read/Write	8	Not supported	One

Figure 6. The Multibus II System Bus has two address spaces each subdivided into two sections

THE SYSTEM BUS

The CPU/Memory bus defined on most buses is inadequate to support "systems-level" requirements so a *SYSTEM SPACE* was added to the definition of the Multibus II System Bus. [A good analogy here, from the software world, is the User/Supervisor Spaces common in advanced operating systems.] This system space is divided into two portions—Interconnect Space to fulfill the initialization, diagnostics and configuration requirements and Message Space to fulfill the standardized communications requirements. Figure 6 shows the four address spaces available on the Multibus II system bus—note that the traditional CPU/Memory space is retained for compatibility with existing buses and to aid migration of existing applications into the Multibus II environment. The system bus is optimized for system space operations but CPU/Memory space operations can perform well in their limited single cycle mode.

Intel's implementation of the Multibus II Parallel System Bus is contained in their VLSI bus interface device, the Message Passing Coprocessor (MPC or 82389), whose functional block diagram is shown in Figure 7. The MPC bus controller is a 70,000 transistor single chip device designed to minimize the board area required by the bus interface circuitry. By standardizing the bus interface in publicly available silicon, all users of the Multibus II standard can look forward to lowering costs and ensured compatibility. This standardization in silicon is similar to Intel's work with IEEE 754 floating point standard implemented in the 8087, 80287 and 80387 components and the IEEE 802.3 Ethernet standard implemented in the 82586 and 82588 components.

The 70,000 transistors which make up the MPC bus controller implement a variety of functions as shown in Figure 8. As seen from Figure 8 most of the MPC bus controller deals with message space, either interrupt messages or data transfer messages or with interconnect space.

Traditional Bus Functions	
Bus Control	4,000
Bus Arbitration	1,000
Dual Port Memory Control	2,000
Off-Board References	1,000
Interrupts	20,000
Advanced Bus Functions	
Parity Generation/Detection	1,000
Interconnect Space	6,000
Built-in-Self-Test	1,000
Message Passing	34,000
Total	70,000

Figure 8. The majority of the 70,000 transistors within the MPC support the System Space functions of the Parallel System Bus

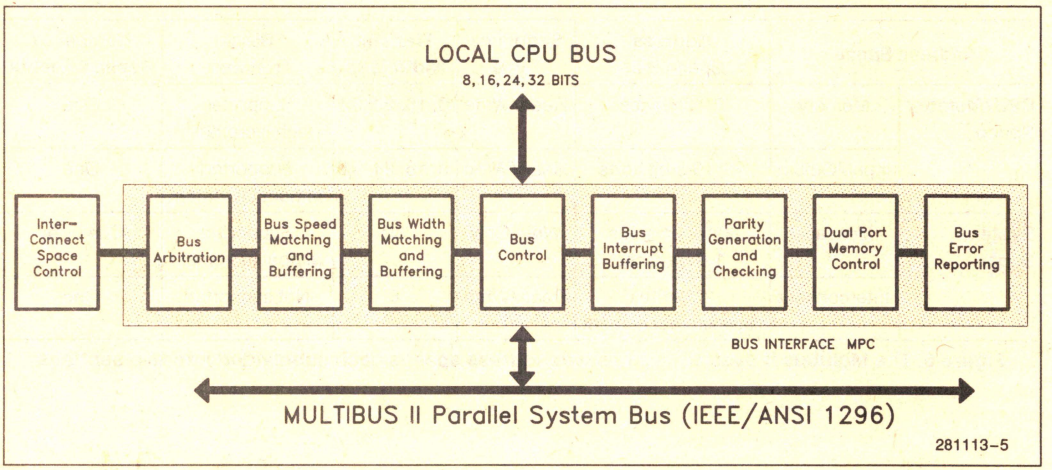


Figure 7. Intel's implementation of the IEEE 1296 specification is cast in silicon as the 82389 component

The MPC bus controller contains almost all of the logic needed to interface any microprocessor to the Parallel System Bus—indeed all of today's popular 32-bit microprocessors are available on Multibus II products. One of the few required external components are the high current bus drivers as shown in Figure 9. Optional external logic to support dual-port memory selection and off-board memory and I/O references may be included if traditional bus functionality is required. All of Intel's Multibus II boards also includes a microcontroller (8751) to implement interconnect space.

The alternate system bus, the Serial System Bus or SSB, is currently defined but is not implemented in silicon. The goal of this bus was to reduce the cost of coupling multiple boards together and it was specified as a 2 Mb/sec serial link. All software interfaces to an SSB chip would be identical to that of the MPC parallel bus controller so NO SOFTWARE CHANGES would be necessary to use the serial system bus. Performance would be much less using this serial system bus but, for many designs, this would be acceptable.

INTERCONNECT SPACE

Interconnect address space is a fundamental part of the IEEE/ANSI 1296 specification and it addresses three major customer requirements: Board identification, initialization, configuration and diagnostics. Interconnect space is implemented as an ordered set of eight-bit registers on long word (32 bit) boundaries—in this way little endian microprocessors such as the 8086 family and the big endian microprocessors such as the 68000 family access the information in an identical manner. One objective of interconnect address space is to allow higher level software to gain information concerning the environment in which it operates, independent of who manufactured the board, the functions it contains, and the card slot it is in. To accomplish this goal, a comprehensive Interconnect Interface Specification which builds upon the concepts introduced within the IEEE/ANSI 1296 specification has been published by Intel Corporation and is available from the Multibus Manufacturers Group.

Board identification registers are read-only locations containing information on the board type, its manufacturer, what components are installed, and other board

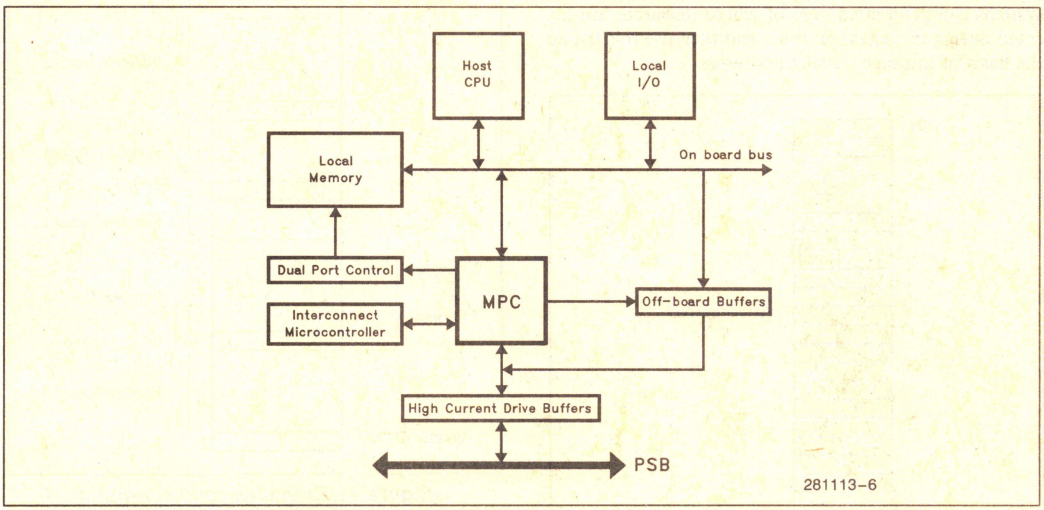


Figure 9. The MPC integrates all of the System Bus functions into a single VLSI component

specific functions. Configuration registers are read/write registers which allow the system software to set and change the configuration of many hardware options. In most cases hard wired jumper options can now be eliminated in favor of software control. Diagnostic registers are used for starting, stopping, and status reporting of self- contained diagnostic routines supplied with each board. These diagnostics are commonly known as Built-in-Self-Tests (BISTs).

Interconnect space is based on the fundamental principle that you can locate boards within a backplane by their physical slot position. This concept, known as geographic addressing, is a very useful tool during system- wide initialization. Each board in the system contains firmware which conforms to a standardized header format as shown in Figure 10. At boot time, the system software will scan the backplane to locate its resources before loading device drivers. This approach eliminates the need for reconfiguring the software every time a new board is introduced to the backplane. It also solves the problem of how to configure multiple controller and processor boards in large multiprocessing systems. Slot independence is achieved by having all boards in the system carry their own initialization and diagnostic functions on-board in firmware. Operating systems can generate a map of where resources are located during initialization time, and then use this list as the basis of message passing addresses.

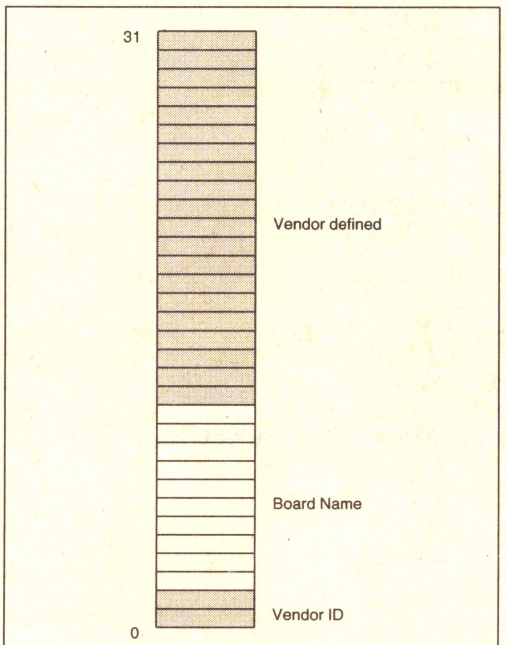


Figure 10. All IEEE/ANSI 1296 compatible boards contain an Interconnect Space Header Record

In addition to the header record, a board manufacturer may also supply additional function records which make other features of the board accessible to the user through interconnect space. An example is shown in Figure 11. Function records begin with a byte specifying the record type, followed by the number of bytes which the function record contains. The data contained in a function record is organized by the manufacturer according to published specifications which accompany the board. Many types of function records have already been defined. Some examples include memory configuration, parity control, serial I/O, and other commonly used functions. If there is no existing record type which adequately describes a given function, new record types can be defined, up to a maximum of 1020 different record types. System software will search for a particular record by starting with register number 32 (end of the header record; start of the first function record), and scanning the record type field, then counting bytes to the next function record until either the correct record is found or an "End of Template" record (hex value 0FFh) is encountered.

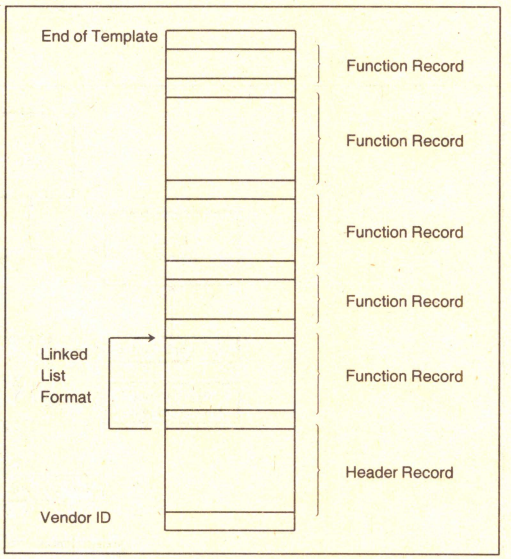


Figure 11. Extended records within Interconnect Space give System Software knowledge of the hardware.

DIAGNOSTIC PHILOSOPHY WITHIN INTERCONNECT SPACE

Intel has taken the usefulness and standardization of interconnect space one step further by embracing a standard diagnostic philosophy. Each intelligent board should have the capability to test itself and report error status in interconnect space if problems exist. There are three occasions when diagnostic testing can be invoked. A subset of the complete on-board diagnostics will be

run during power-on initialization system testing, and more extensive testing can be invoked from an operators console. Following power-on, most boards will go through a series of initialization checks, where the basic functioning of the MPC bus controller and microcontroller is verified. These checks are followed by a power-on test suite which is controlled automatically by each local microprocessor. If a hardware failure is detected at this point, a yellow LED on the front panel will illuminate so that the failing module can easily be identified and replaced by an operator, additionally test results are posted in interconnect space to be read across the backplane. Note that a CPU board when scanning interconnect space can now discover the operational status of boards in the backplane as well as their identity.

System level diagnostics are run to make sure a single board does not corrupt the rest of the system's operation. If a board is not "well enough" to run in a system environment it will be disabled (CPU shutdown) by the microcontroller.

If further testing is desired, extended diagnostics can be invoked by placing a diagnostic request in the BIST registers of interconnect space. Usually one board will operate as a Master Test Handler, and will request services from other boards in the system which function as Slaves while under test. A menu of available tests is accessible via interconnect space. This test philosophy can be applied on-site by the end-user or service representative, or remotely executed via modem from a regional repair center. In most cases, downtime can be minimized by sending out a replacement board, thus avoiding an expensive repair call.

The firmware content of Multibus II boards is much greater than on previous industry standard buses. In addition to the 8751 microcontroller, there are likely to be EPROMs on board which contain the extended diagnostics, test handlers, reset initialization sequencing, debug monitors, and numerous other functions. The location of diagnostic firmware on a board will depend on the complexity of the code and the speed at which it runs. For simple replier agents, it may be that the onboard EPROM of the 8751 microcontroller contains enough program store for rudimentary diagnostic functions as well as the interconnect core firmware. In contrast, most requestor/replier boards (those capable of becoming bus masters) are more complex, and most diagnostic code is run by the microprocessor from onboard EPROM. In this case, the 8751 serves primarily as the communications interface for diagnostics.

INTERCONNECT SPACE-THE MANUFACTURERS PERSPECTIVE

From the perspective of a board designer, interconnect is a mixed blessing. The board manufacturer is certain to enjoy the benefits of reduced support costs, easier fault isolation in field repairs, and enhanced customer satisfaction, but these advantages do not come for free.

One would anticipate longer development times, increased parts count on-board, and configuration in firmware to increase the amount of effort it takes to prepare a Multibus II board for market. Indeed this is so. In order to minimize this development time Intel has produced an Applications Note which details the steps and discusses the options available for a full featured interconnect space implementation. The core microcontroller code is also provided on a DOS diskette and is designed to user extensible. It is now straightforward to add these advanced capabilities to any Multibus II board design.

THE MESSAGE PASSING MECHANISM

While the previously described features make more reliable systems easier to build using the Multibus II standard, it is the innovative message passing scheme that gives the parallel system bus its high performance in a multiple microprocessor application. The underlying theory behind message passing is simple—it decouples activities between the host microprocessor's local bus and the system bus. This decoupled-bus approach provides two major advantages. First, it allows increasing parallelism of operation—resources that would otherwise be held in traditional wait states while arbitration occurs are freed, and second, one bus bandwidth does not limit the transfer rate of another. The local microprocessor bus and the system bus can perform full speed synchronous transfers independently and concurrently. The decoupling is achieved within the MPC bus controller using high speed FIFO circuitry as shown in Figure 12.

Nine 32 byte FIFO's are integrated into the MPC bus controller. Five of them are used for interrupt messages (one transmit and four receive) and four are used for the transfer of data blocks (two transmit and two receive). To understand the impact of message passing, let us consider a simple example of transferring 1K block of data from CPU A to CPU B as shown in Figure 13. We will first use a shared memory method and then a message passing method.

To use a concrete example lets assume that A is a 186 based board and can transfer data at 1 MB/sec and B is a i386 based board that can receive data at 10 MB/sec. We will ignore DMA controller setup. DMA controller A will put a destination address onto the system bus and the address decode logic on board B will respond. We wait for the address to propagate through the dual-port controller on board B and then wait for the access time of the memory on board B. Data is

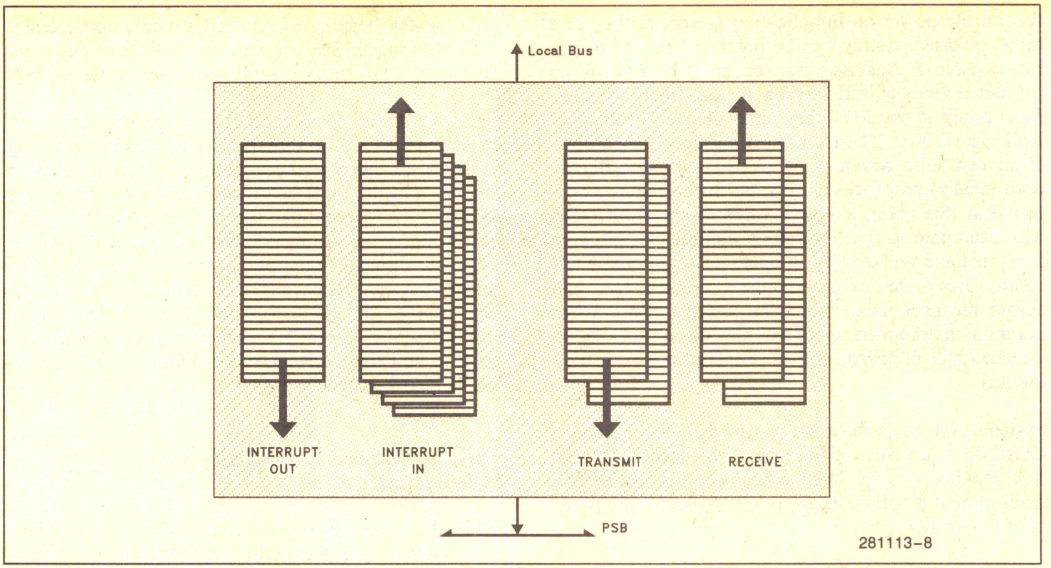


Figure 12. Decoupling of the local bus from the system bus is achieved with nine very high speed FIFOs

transferred and once accepted by board B a ready signal will be generated and DMA controller A will move on and generate the next address. This address-wait-data cycle repeats until the full 1 KB of data is transferred. The overall speed of the transfer will be 1 MB/sec (the slower of the two boards) so it will take 1 msec to transfer the complete 1K buffer. If the system bus was required by an alternate CPU then the current data transfer would be delayed or the alternate CPU would have to wait.

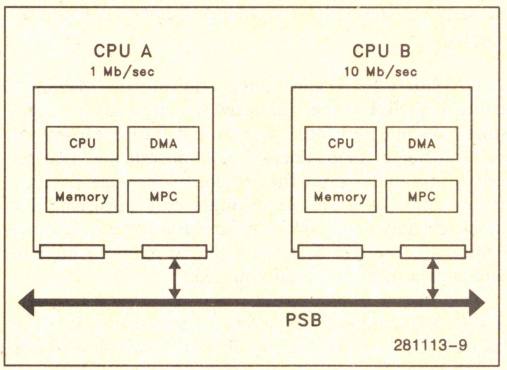


Figure 13. Let us move a 1 Kbyte block of data from Board A to Board B

Now lets consider the message passing case. This time we have to set up both DMA controllers. CPU A could probably transfer data faster than 1 MB/sec into a local I/O port (the MPC bus controller) but we will ignore this potential performance improvement. The speed of this transfer will still be 1 MB/sec, the speed of the slower board, and the total transfer time will still be 1 msec. What did we gain then for the overhead of setting up two DMA controllers?

Let us look in detail at what is happening inside the MPC bus controller. Figure 14 shows a fragment of each board with different areas of each MPC bus controller highlighted. Data is being DMA'ed into MPC-A at 1 MB/sec and flows into one of the transmit FIFO-pairs. Once 32 bytes have been received the MPC automatically switches to the alternate transmit FIFO and starts to fill that. The full transmit FIFO empties across the system bus into a receive FIFO in MPC-B. This transfer of a 32 byte packet occurs at the full bus bandwidth of 40 MB/sec. A data packet has a two clock cycle header which describes the source, destination and type of this packet which reduces the effective data transfer rate to 32 MB/sec. The packet therefore takes 1 microsecond to pass between the two MPC bus controllers. Bus arbitration is done in parallel with the packet transfer so this does not add to the transit time. Once the packet is inside of MPC-B then DMA-B empties its receive FIFO at 10 MB/sec.

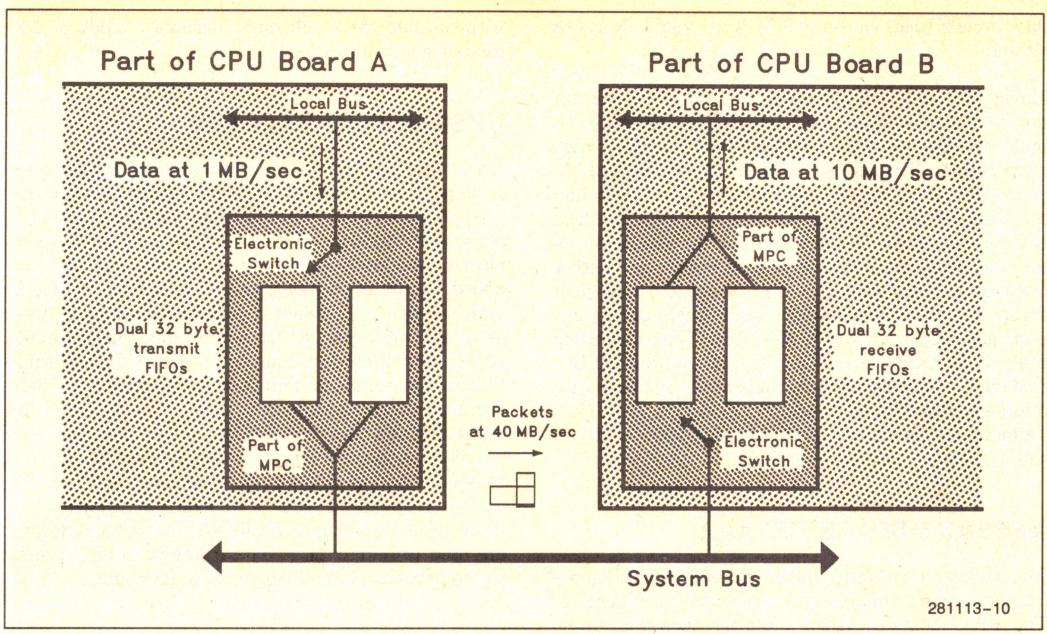


Figure 14. Looking closely at the message based data transfer mechanism

The transmit FIFO-pair of MPC-A alternate between filling from local memory and emptying into MPC-B until the full 1K of data has been transferred. No programming, save the initial setup, is required. If we look at the system bus activity we see that 1 microsecond packets are being transferred at 32 microsecond intervals—the bus is only busy for 3% of the total data transfer. We have gained 97% bus availability. Compare this with the 0% bus availability in the shared memory case. Message passing frees up system bus bandwidth to enable many other single board computer pairs to interchange data at no loss in performance. In a multiple microprocessor application the most precious resource will be system bus bandwidth and the Multibus II message passing scheme gives you more, much more.

We gain a lot more than system bus bandwidth using message passing. Not that CPU A transferred data from its local memory into a local I/O port (the MPC bus controller). CPU A did not have to understand the memory layout or restrictions of memory on CPU B—this also allowed CPU B to do its own memory management and buffer allocations. Similarly CPU B has no concern over how CPU A does its memory management. We have isolated the data away from known memory locations and do not have to deal with semaphore flags or similar mechanisms. This simplifying step make intercommunicating with multiple microprocessors as straight forward as communication with a single microprocessor. This isolation of concerns regarding the local environments of each board, through the use of a standardized data transfer mechanism, is especially important in the general case where each board is running a different operating system (probably on a different microprocessor). A real time operating system can now simply exchange data with, say, UNIX* using this standardized message passing mechanism. Message passing also standardizes inter-CPU signalling since interrupts are special TYPEs of packet (more later).

This short explanation has over-simplified the transfer—some setup is required so that the sending MPC bus controller knows the message address of the receiving MPC bus controller, etc. This overhead is more than compensated for by the ignored increase in local transfer data rates. I also simplified the issue by having a receiving board much faster than the transmitting board (10 MB/sec vs. 1 MB/sec)—if I had transferred data in the opposite direction (from B to A) then MPC-A would have rejected some packets because its receive FIFOs would be full and caused MPC B to retry some data transfers. No data would ever be lost but bus activity would have increased. The MPC bus controller uses a logarithmic backoff algorithm on retries so the bus activity increase would not be excessive. Alternately MPC B could be preprogrammed to use a lower packet

duty cycle if I had known that MPC A would always be slower.

Having the underlying architectural support to permit multi-CPU solution is, of course, only the first step. To build systems we need software. Intel, working with other vendors, has defined a Transport Protocol specification above the MPC bus controller which provides services such as large block transfers and acknowledged transactions. Data fragmentation at the sender or receiver is detailed sot that large data buffers are neither assumed nor required. The implementation is efficient across all CPU architectures; indeed, Intel has supplied implementations on the iRMX Real Time Operating System, the iRMK Real Time Kernel and the UNIX* System V operating systems; these are compatible with offerings from Digital Research (FLEXOS*, VRTX*, Tadpole Technologies UNIX68K, MS-DOS*, Microsoft Windows*, plus many more).

MESSAGE SPACE DETAILS

The MPC bus controller introduces a hardware recognized data type called a packet as shown in Figure 15. The MPC contains FIFO circuitry such that these packets may be moved very efficiently between MPCs—data is moved on subsequent clock edges of the 10 MHz synchronous bus; this defines the maximum bus occupancy of a packet to be one microsecond. Each MPC bus controller has an address in message space and these are used in the message header (source and destination fields).

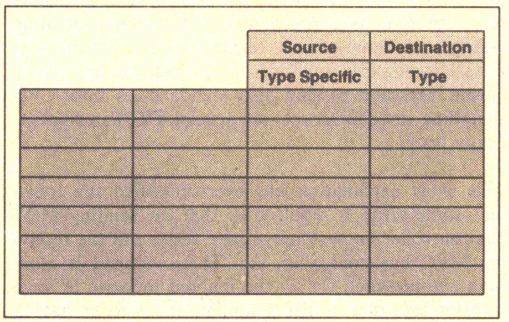


Figure 15. The MPC bus controller introduces a hardware-recognized data type called a packet

Seven different packet types are currently defined and are summarized in Figure 16. These divide into two categories; unsolicited, or interrupt packets and solicited, or data transfer packets. The data fields within a packet are user defined and the length may vary from zero to a maximum of 32 bytes (28 for an unsolicited packet) in four-byte increments. Note that a packet with no data bytes will only consume 2 clock or 200 nsec of system bus time.

UNSOLICITED PACKETS

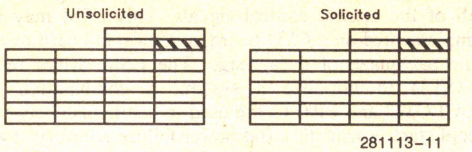
Unsolicited packets, as the name implies, are always a surprise to the MPC bus controller. Their arrival is unpredictable so each MPC has four FIFOs in which it can queue unsolicited packets. These packets are equivalent to the interrupts of a conventional bus with the added feature of having up to 28 data bytes provided with the signal. Unsolicited packets can be thought of as virtual interrupts. Where as traditional bus are limited to 5.8 or 11 hardware interrupts across the bus, Multibus II can handle unlimited or 255^{21} interrupts.There are five different unsolicited packets: two are used for interrupts and three are used to set up solicited data transfers. A general interrupt request may be sent between any pair of single board computers and a broadcast interrupt may be sent to all boards in a system. The three special interrupts, Buffer Request/Grand/Reject are used to initiate large data transfers (solicited messages) between pairs of single board computers.

SOLICITED PACKETS

Solicited packets are never a surprise since their arrival is negotiated—the receiving MPC bus controller knows what to do with them. These packets are used for the transfer of data from one board to another and the transfer is set up using unsolicited packets. To summarize the operation of solicited packets the MPCs cooperate in the moving of blocks of data between boards, they break the data into 32 byte packets, send them across the bus, and reassemble the data transparently to the sending/receiving boards. All operations such as packetization, bus arbitration, error detecting and recovery is handled by the MPC bus controllers—this is done transparently to the local microprocessors. Key to system performance is the packetization of data on the system bus which limits the maximum bus occupancy to one microsecond.

BUS ARBITRATION GUARANTEES LOW BUS LATENCY

The Multibus II system bus uses a distributed arbitration scheme as shown in Figure 17. Each board that requires access to the system bus contains the circuitry of Figure 17 (note that this is contained within the MPC bus controller) and no active components are required on the backplane. The MPC bus controller uses a software assigned identifier to request the bus and the arbitration circuitry will indicate that the MPC is preparing for a bus cycle. The MPC supports two arbitration algorithms, fairness and high priority.



- Intelligent Interrupts
- Used for signalling
- User data 0 to 20 bytes
- Types
  - — 00 General interrupt
  - — 01 Broadcast interrupt
  - — 36 Buffer Request
  - — 52 Buffer Reject
  - — 53 Buffer Grant

- Negotiated arrival
- Used for data transfer
- User data up to 16 MB
- Types
  - — 62 Data Packet
  - — 63 Last Data Packet

Figure 16. There are seven type of packet subdivided into two categories

The fairness mode is used for data transfers and is "polite". If the bus is being used, the MPC will wait before requesting use of the bus; once the bus is not busy the MPC will request the bus and will wait for it to be granted; once the MPC uses the bus it will not request it again until all other requesters have used the bus. "Parking" on the bus is permitted—if no other board has requested the bus since the last time that this board accessed the bus (remember that the bus is continually monitored) then it may access the bus directly without executing an arbitration cycle. These algorithms assure that a single board cannot monopolize the bus and keep others from using it. Remember that each MPC will only use the bus for a maximum of one microsecond and since the arbitration is being resolved in parallel there are no wasted clocks as bus ownership is transferred; all transfers operate back-to-back.

THE SYSTEM BUS HAS DETERMINISTIC INTERRUPT LATENCY

The high priority mode is used for interrupts and is "impatient". The MPC bus controller, when in this mode, will "barge in" on an arbitration cycle and be guaranteed the next access to the bus. The MPC bus controller can set up interrupt packets specifically to operate in high-priority mode so these will only have a one microsecond latency to access the bus. In the rare instance that two MPCs try to initiate an interrupt packet within the same one microsecond window the highest priority board will be granted the bus and the other board will have to wait a maximum of two microseconds for its bus access.

Interrupt packets and data transfer packets interleave on the bus (actually interleave within a single MPC too) with preference always going to the interrupt packet. A single MPC bus controller will operate with interrupt packets in high priority mode and data transfer packets in fairness mode; this will ensure that interrupt packets have a deterministic bus transit time of 1 μsec (2 μsec sometimes and 20 μsec conceivable worse case when all 21 MPCs try to initiate an interrupt packet within the same one microsecond window, a very, very rare occurrence).

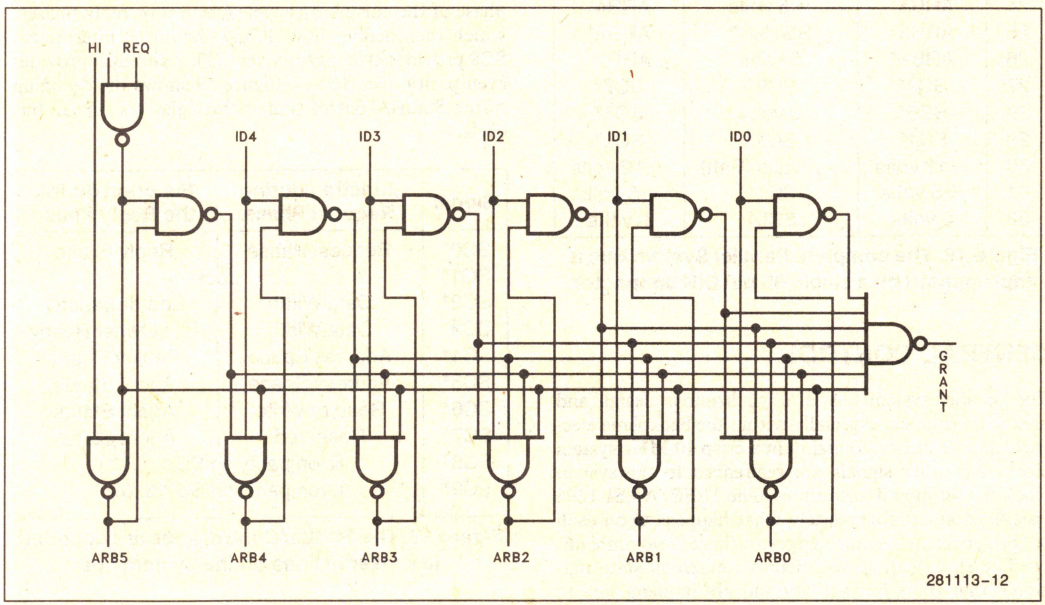


Figure 17. The Multibus II Parallel System Bus uses a distributed arbitration scheme

SYSTEM BUS CHARACTERISTICS

Figure 18 shows the pinout of the P1 connector. The signals can be classified into five groups; 1. Central control, 2. Address/data, 3. System control, 4. Arbitration, and 5. Power.

Pin	Row A	Row B	Row C
1	0 volts	PROT*	0 volts
2	+5	DCLOW*	+5 volts
3	+12 volts	+5 volts (Batt)	+12 volts
4	0 volts	SDA	BCLK*
5	TIMOUT*	SDB	0 volts
6	LACHn*	0 volts	CCLK*
7	AD0*	AD1*	0 volts
8	AD2*	0 volts	AD3*
9	AD4*	AD5*	AD6*
10	AD7*	+5 volts	PAR0*
11	AD8*	AD9*	AD10*
12	AD11*	+5 volts	AD12*
13	AD13*	AD14*	AD15*
14	PAR1*	0 volts	AD16*
15	AD17*	AD18*	AD19*
16	AD20*	0 volts	AD21*
17	AD22*	AD23*	PAR2*
18	AD24*	0 volts	AD25*
19	AD26*	AD27*	AD28*
20	AD29*	0 volts	AD30*
21	AD31*	Reserved	PAR3*
22	+5 volts	+5 volts	Reserved
23	BREQ*	RST*	BUSERR*
24	ARB5*	+5 volts	ARB4*
25	ARB3*	RSTNC*	ARB2*
26	ARB1*	0 volts	ARB0*
27	SC9*	SC8*	SC7*
28	SC6*	0 volts	SC5*
29	SC4*	SC3*	SC2*
30	−12 volts	+5 volts (Batt)	−12 volts
31	+5 volts	SC1*	+5 volts
32	0 volts	SC0*	0 volts

Figure 18. The complete Parallel System Bus is implemented on a single 96 pin DIN connector.

CENTRAL CONTROL

The parallel system bus is a synchronous design and great care is taken, especially within the backplane electrical specifications, to maintain a crisp 10 MHz system clock. All other signals are referenced to the system clock for setup and hold times. The IEEE/ANSI 1296 specification details precisely what happens upon each of the synchronous clock edges so there is no ambiguity. The specification also details numerous state machines that track bus activity and are implemented to guarantee compatibility.

A central services module (or CSM) in slot 0 generates all of the central control signals. This CSM may be implemented on a CPU board, a dedicated board or on the backplane of a cardcage. The CSM drives reset (RST*) to initialize a system; a combination of DCLOW* and PROT* are used to distinguish between cold start, warm start and power failure recovery. Two system clocks are generated, BLCK* at 10 MHz and CLCK* at 20 MHz. RSTNC* and LACHn are used for advanced facilities within the bus and their description is deferred.

SDA and SDB are reserved for a serial system bus (currently being investigated by the IEEE) and there are two pins reserved for future use.

ADDRESS/DATA

The Multibus II parallel system bus is a full 32 bits (AD0..31*) with byte paraty (PAR0..3*). The system control lines will define when address information or data is contained upon these multiplexed lines. Note that all transfers are parity checked and, in the case of message packets, the MPC bus controller will retry an operation that failed due to a parity error. If, after sixteen tries, the error is not recoverable, the MPC bus controller will interrupt its host microprocessor to ask for assistance.

SYSTEM CONTROL

Ten lines (SC0..9*) are used for system control and their functions are multiplexed too. SC0* defines the phase of the current bus cycle (request or reply phases) which then defines how SC1..7 should be interpreted. SC8 provides even parity over SC0..3 and SC9 provides even parity over SC4..7. Figure 19 shows the decoding of the Status/Control signals throughout a typical bus cycle.

Signal	Function during Request Phase	Function during the Reply Phase
SC0*	Request Phase	Reply Phase
SC1*	Lock	
SC2*	Data Width	End-of-transfer
SC3*	Data Width	Bus Owner Ready
SC4*	Address Space	Replier Ready
SC5*	Address Space	Agent Status
SC6*	Read or Write	Agent Status
SC7*	Reserved	Agent Status
SC8*	Even parity on SC<7..4>*	
SC9*	Even parity on SC<3..0>*	

Figure 19. The Status/Control lines are encoded to preserve lines on the system bus

ARBITRATION

All boards request use of the bus through a common bus request line BREQ*. A distributed arbitration scheme is defined which grants the bus to the numerically highest requesting board as identified on lines ARB0..5. Two arbitration algorithms are supported: fairness, which gives each board an even portion of the available bus bandwidth, and priority, which permits a high priority request (such as an interrupt) to be guaranteed the next access to the system bus.

POWER

There are ample power and ground lines defined and these are spread over the length of the P1 connector to minimize ground shift and other problems.

TYPICAL BUS CYCLE

The parallel system bus is particularly easy to interface to. This section will cover the sequencing of a typical REPLIER interface as an illustration of the bus timing. The IEEE/ANSI 1296 specification details numerous state machines that track bus activity and are implemented to guarantee compatibility. An I/O replier need only implement a single "Replying Agent" state machine. This is shown in Figure 3.5-5 in the IEEE/ANSI 1296 standard and repeated here in Figure 20 for reference. Remember that an application CPU (a REQUESTOR) will start the cycle that the REPLIER will respond to.

In order to progress quickly through this discussion, an assumption that the requestor always issues valid requests will be made. Error handling for invalid requests will be added later. Figure 21 summarizes the design task. The logic required to map the multiple signals and protocols from the Multibus II parallel system bus into the simple read strobe, write strobe and chip select of an I/O device must be designed. In this example features will be kept at a design minimum but all essential circuitry will be discussed in detail.

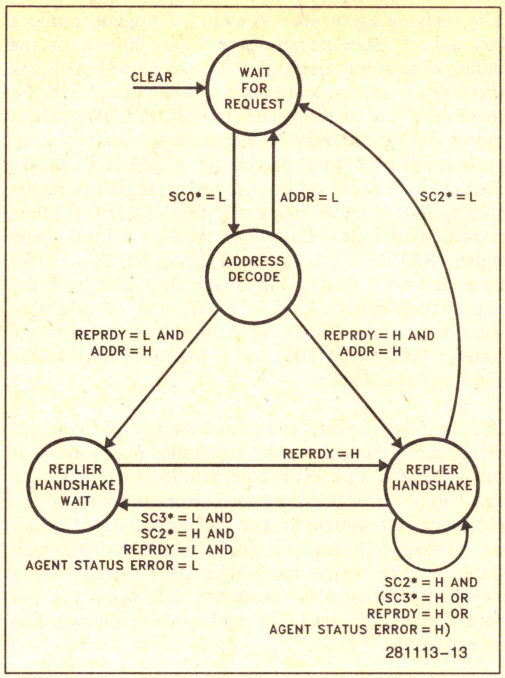


Figure 20. The IEEE/ANSI 1296 Specification details numerous state machines. A replier is shown here.

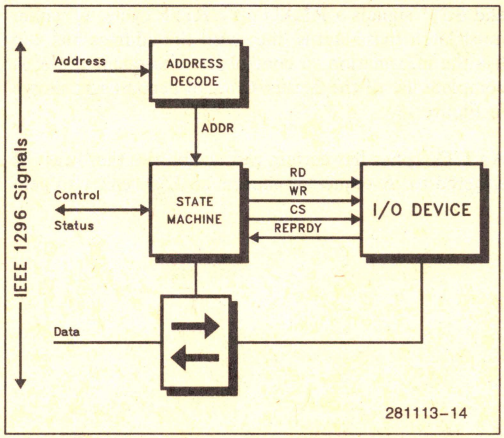


Figure 21. The design of a REPLIER is fundamentally a bus monitor.

The replying agent state machine is fundamentally a bus monitor. State transitions in Figure 20 occur at the falling edge of bus clock. The state machine remains in the wait-for-request state until it detects the start of a requestor cycle on the system bus (SC0* LOW) then it moves into an address decode state. If this requestor cycle is not ours (local decode signal ADDR is LOW) then return to the wait-for-request state. If the requestor cycle is detected as ours (ADDR is HIGH) then transition to a new state controlled by a local ready signal (REPRDY). If not ready (REPRDY is LOW) then wait until ready. Once ready then wait until the requestor is ready (SC3* is LOW) and provide/consume valid data. Check to see if this is a multi-byte transfer (SC2* is HIGH) and if it is not return to the wait-for-request state.

If a multi-byte transfer is detected then decide to accept or to ignore the data in the remainder of the cycle. If the additional data cannot be handled then signal an agent status error (Continuation error) and wait for the requestor to terminate the cycle. If a multi-byte transfer can be supported then oscillate between the replier wait state and the replier handshake state where data is strobed. Eventually the requestor will signal the last data element (SC2 set LOW) and return to the wait-for-request state.

At the start of each requestor cycle that status lines (SC1* through SC6*) detail the type of cycle; SC1* signals a locked transfer, SC2* and SC3* encode the data width, SC4* and SC5* encode the address space and SC6* signals a READ or WRITE cycle. A replier must latch these status lines with the address bus and use the information to control its subsequent cycle. A complete list of the Status/Control decoding is shown in Figure 19.

An I/O replier has certain responsibilities that must be adhered to. A requestor expects an I/O replier to generate status information and to signal when ready so that the requestor may proceed with the cycle. The cycle will only terminate once both requestor and replier have signalled that they are ready (the IEEE/ANSI 1296 includes a time out feature which prevents the bus from hanging if both ready signals are not generated). A replier drives SC4* LOW to indicate READY and status information is driven on lines SC5* through SC7*; SC8* must also be driven and identifies parity across lines SC4* through SC7*. If a replier is supplying data to a requestor then correct data parity must also be driven onto the system bus.

SUMMARY

The Multibus II Parallel System Bus was *DESIGNED* to implement all of the "systems features" of a single board computer based system. The bus does have some CPU/Memory attributes but these were only included for compatibility and to aid migration into the Multibus II environment—comparing these CPU/Memory features in isolation with those of other buses is a complete disservice to the Multibus II architecture and misses the complete design goals and motivation set forth for this standard.

The silicon revolution forced the design of the Multibus II Parallel System Bus—technology was advancing faster than our abilities to use it so we had to find new implementation strategies to benefit from these advances. Functional partitioning was chosen as the vehicle to embrace the technology; by partitioning the problems into smaller and smaller sub-problems we reach a point where the sub-problems are implementable. The Multibus II consortium chose this path and executed with precision; transistors were applied at strategic points to simplify implementations and encourage ease-of-use. The Multibus II architecture is completely defined, documented and available.

intel®

AP-422

APPLICATION NOTE

Designing a Central Services Module for MULTIBUS II

5

JORY RADKE
MODULES DEVELOPMENT ENGINEER

September 1993

Order Number: 280634-001

CONTENTS

PAGE

1.0 INTRODUCTION 5-19

2.0 CSM FUNCTIONS AND PSB SIGNALS 5-19

2.1 CSM Functions 5-19

2.2 PSB Signal Used by the CSM 5-19

3.0 FUNCTIONAL OVERVIEW 5-19

3.1 Clock Generator (CLKGEN) Function 5-19

3.2 Reset Generator (RSTGEN) Function 5-21

3.3 Timeout Generator (TOGEN) 5-26

3.3.1 Transfer Cycle Timeout 5-26

3.3.2 Bus Ownership Timeout 5-26

3.4 Cardslot and Arbitration ID Assignment Generator (IDGEN) Function 5-28

CONTENTS

PAGE

3.4.1 Eight Count BCLK* Delay 5-28

3.4.2 Sequencer Logic 5-30

4.0 DESIGN EXAMPLE 5-30

4.1 PSB Buffer Interface (Sheet 1) 5-31

4.2 Clock Generator (CLKGEN) (Sheet 2) 5-31

4.3 Reset Generator (RSTGEN) (Sheet 2) 5-32

4.4 Timeout Generator (TOGEN) (Sheet 3) 5-32

4.5 Cardslot and Arbitration ID Generator (IDGEN) 5-32

5.0 DESIGN CONSIDERATIONS 5-44

5.1 Cardslot 0 Detection 5-44

5.2 System Diagnostic Options 5-45

5.3 CSM Functions on the Backplane 5-45

PURPOSE

This paper describes and presents methods for implementing the functions provided by the Central Services Module, as defined in the IEEE 1296 specification, and is intended to assist the system designer in understanding and effecting these functions. Function options and other design considerations are discussed. It is assumed the reader is familiar with the terms and definitions used in the IEEE 1296 specification and with basic logic design principles.

1.0 INTRODUCTION

The IEEE 1296 specification, based on the Intel MULTIBUS II bus architecture, defines certain general system-wide functions to be provided by a Central Services Module (CSM). These system-wide functions include power-on and power-fail reset sequences, clock generation, bus timeout detection and signal generation, and the assignment of cardslot and arbitration identification (ID) to each board (agent). The communication path between the CSM and the other agents in the system environment is via the Parallel System Bus (PSB) interface.

The implementation of the defined CSM ensures uniformity in providing a single source for those system-wide functions required in an open-bus architecture, such as that established in the IEEE 1296 specification. Centralizing system-wide functions reduces system cost and frees board area for other functions since only one board in the system need contain the CSM logic. The IEEE 1296 specification stipulates that only the agent in cardslot 0 contain the active CSM functions although other system agents may contain CSM functions.

2.0 CSM FUNCTIONS AND PSB SIGNALS

The following paragraphs identify and briefly describe the system-level services and functions supplied by the CSM and the PSB signals generated, monitored or used to implement these services. An asterisk following the signal name indicates that the particular signal or group of signals are active when at their electrical low.

2.1 CSM Functions

The IEEE 1296 specification defines the minimum required functions of a CSM as:

- Generation of system clock signals
- Generation of reset sequences for both cold and warm start and power failure indication
- Cardslot and arbitration ID initialization
- Timeout signal generation for PSB data transfer cycles.

Depending on system requirements, the CSM may additionally provide:

- Power-fail recovery reset
- Bus ownership timeout.

Other system-wide resources, such as a time of day clock or interface to another bus system, may be conveniently implemented with the CSM on the same PSB agent. We shall see that the CSM functions require very little board area to implement.

2.2 PSB Signal Used by the CSM

The CSM utilizes signals from each of the five signal groups defined in the IEEE 1296 specification. These signals are identified and their use by the CSM is described briefly in Table 2-1.

3.0 FUNCTIONAL OVERVIEW

The following sections discuss how to add the CSM functions to a PSB agent. The agent could contain only the CSM and interconnect relier modules and additional functional modules as well. The design example provided in section 4, (excepting the PSB buffers), requires less than six percent of the area on a standard MULTIBUS II board. The CSM module cannot be added to agents which employ Intel's Message Passing Coprocessor, due to the current and capacitive loading requirements of the PSB signals in Table 2-1 which the MPC drives directly.

This paragraph provides a functional overview of the design and discusses signal requirements. A detailed design example is illustrated and discussed in paragraph 4. Additional design considerations are described in paragraph 5.

Functionally partitioning the CSM functions results in the block diagram shown in Figure 3-1. The signal terminations identified on the righthand side of the diagram are the actual PSB pin assignments identified in the IEEE 1296 specification.

3.1 Clock Generator (CLKGEN) Function

Listed in Table 3-1 and depicted in Figure 3-2 are the timing relationships between the **BCLK*** and **CCLK*** signals as specified in the IEEE 1296 specification. The circuits used to develop and supply the **BCLK*** and **CCLK*** signals must guarantee t2 (high time), t4 (low time), and t5 (period). The circuits must also guarantee t6 (clock-to-clock) and the correct phase relationship between clock signals. Signal parameters t1 (rise time) and t3 (fall time) must be met by the buffer device driving the clock signals onto the PSB bus interface.

Table 2-1. PSB Signals Used by the CSM

Signal		CSM Function
Name	Group	
BREQ*	Arbitration Cycle	Bus Request. All agents that require access to the PSB assert the **BREQ*** signal. The CSM monitors this signal as part of its bus timeout function.
ARB<5..0>*	Arbitration Cycle	Arbitration lines. The CSM uses these lines during a reset sequence to assign a cardslot ID and an arbitration ID to each agent in the system.
AD<20..1>*	Address/Data Bus	Address/Data lines. See Figure 2-1. Each Address/Data line is connected to the **LACHn*** pin of a cardslot. The **LACHn*** signal is used to latch the cardslot and arbitration IDs to each agent (except cardslot 00) during a reset sequence.
SC<4..2, 0>*	System Control	System Control lines. The CSM monitors these control signals between agents to sense bus timeout during data transfer cycles.
TIMOUT*	Exception Cycle	Bus Timeout. **TIMOUT*** is asserted by the CSM to signal that an agent is taking too much time to respond to a handshake.
BUSERR*	Exception Cycle	Bus Error. An agent activates **BUSERR*** to indicate its detection of a data integrity problem during a transfer. The CSM monitors this signal as part of its bus timeout function.
BCLK*	Central Control	10 MHz Bus Clock. Driven only by the CSM to provide all system timing references.
CCLK*	Central Control	20 MHz Central Clock. Driven only by the CSM as an auxiliary clock for use as an additional timing reference among bus agents.
RST*	Central Control	Reset. Driven only by the CSM as a system-level initialization signal.
DCLOW*	Central Control	DC Power Low. Driven only by the CSM as a warning to system agents of an imminent power failure. Part of the CSM reset generation function.
PROT*	Central Control	Protect. Driven only by the CSM during power-fail sequences. Part of the reset generation function.

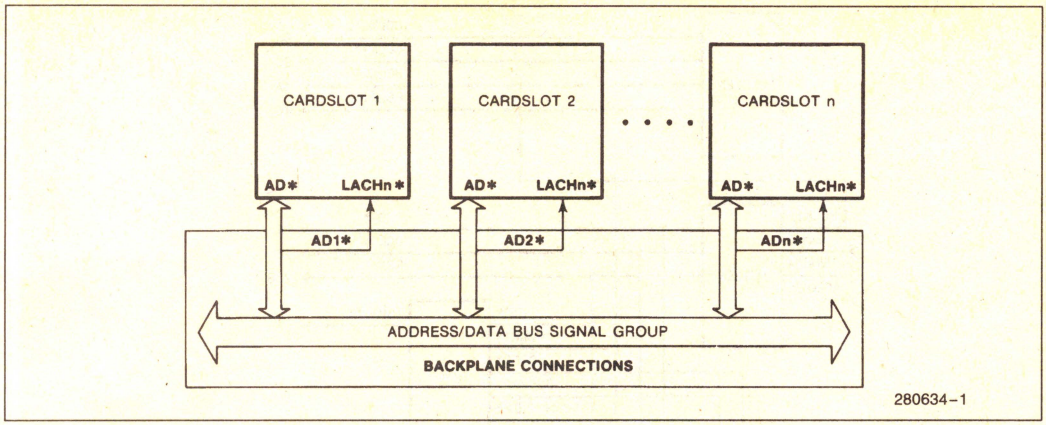


Figure 2-1. Backplane Connection of LACHn*

NOTE:
The **BCLK2*** and **CCLK2*** signals are only required for backplanes containing more than 12 cardslots.

The CSM clock functions can be implemented by use of a crystal oscillator, frequency divider, and two or more bus drivers.

3.2 Reset Generator (RSTGEN) Function

The IEEE 1296 specification defines three types of reset sequences for the CSM: cold, warm and recovery. The **RST***, **DCLOW*** and **PROT*** signals are used to encode the reset type. The **DCLOW*** and **PROT*** signals are defined as being asynchronous while the **RST*** signal is defined as being synchronous.

The **ACLOW*** input is only from the power supply in systems supported by battery backup (VBB) and is required for power fail and recovery resets. The **WARM** and **COLD** inputs represent user-controlled signals for use in generating warm or cold resets. They might be supplied from a system front panel or via a status register in the agent's interconnect space. **BCLK1*** is an input to clock the synchronous **RST*** signal.

Tables 3-2 through 3-4 and Figures 3-3 through 3-5 list and depict the timing specifications for the cold, warm and power-fail recovery resets, respectively.

There are various system and user defined parameters beyond the scope of this article which can be added to the design and implementation of the RSTGEN function. Exploring the flexibility presented by such additional factors as the characteristics of the **ACLOW*** signal, whether or not to support battery backup, the ramp-up time of the power supply, the number of front panel or user inputs, or which options to permit when multiple resets occur simultaneously are left for the design engineer's consideration.

The RSTGEN function described above can be implemented using voltage monitors, timers and basic control logic.

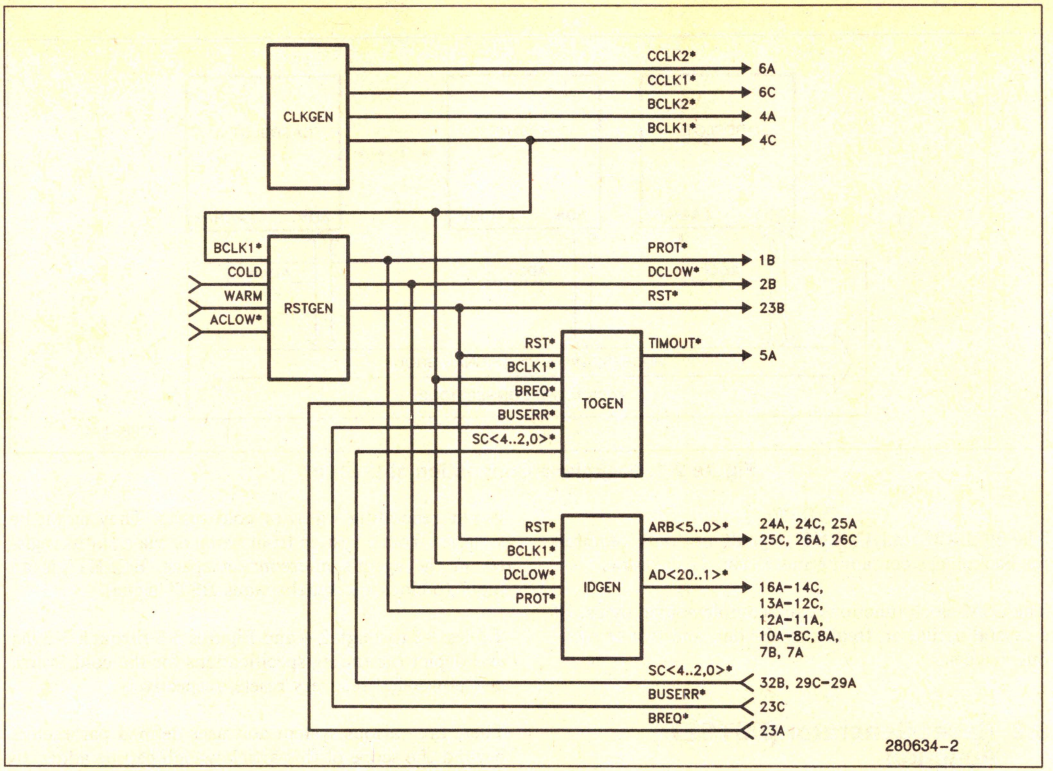


Figure 3-1. CSM Functional Block Diagram

Table 3-1. CSM Clock Timing Specification

Symbol	Parameter	BCLK*		CCLK*		Units
		Min	Max	Min	Max	
t_1	Rise Time		2.0		2.0	ns
t_2	High Time	48.0	52.1	23.0	27.0	ns
t_3	Fall Time		2.0		2.0	ns
t_4	Low Time	48.0	52.1	23.0	27.05	ns
t_5	Period	99.9	100.1	49.95	50.05	ns
t_6	Clock-to-Clock	0	+10			ns

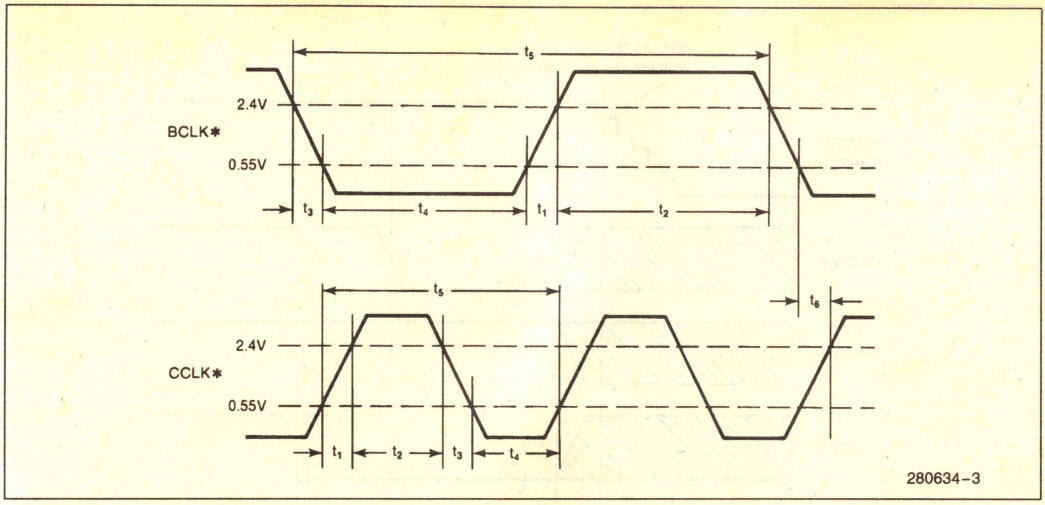


Figure 3-2. Clock Timing Relationships at CSM Connector P1

Table 3-2. CSM Cold Reset Timing Specifications

Symbol	Parameter	Min	Max	Units
t_1	DC Power Setup to **DCLOW***		1.0	ms
t_2	Cold Reset Duration	2.5		ms
t_3	Warm Reset Duration	50.0		ms

5

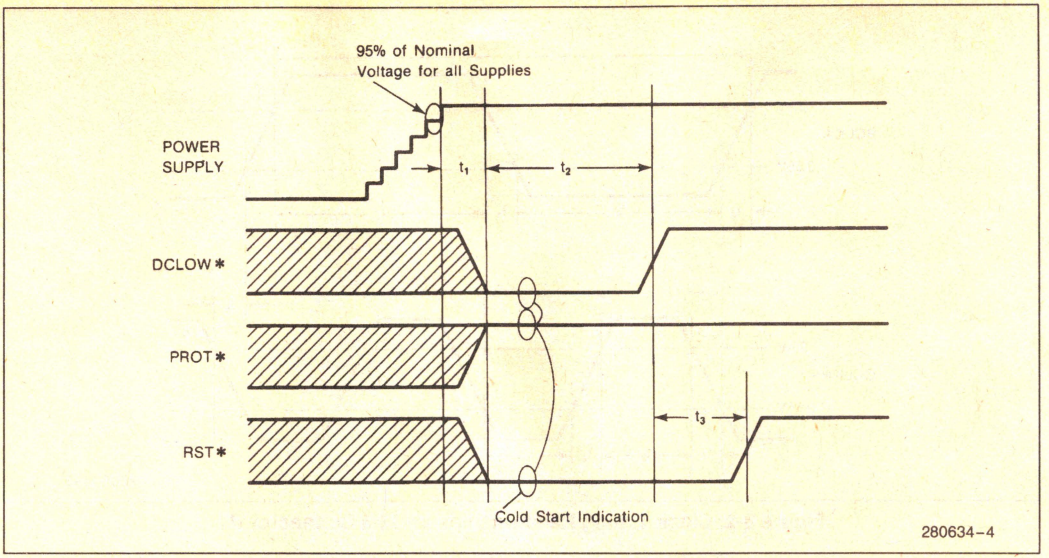


Figure 3-3. Cold Reset Timing on the PSB

Table 3-3. CSM Warm Reset Timing Specifications

Symbol	Parameter	Min	Max	Units
t_1	**RST*** Pulse Width	50.0		ms

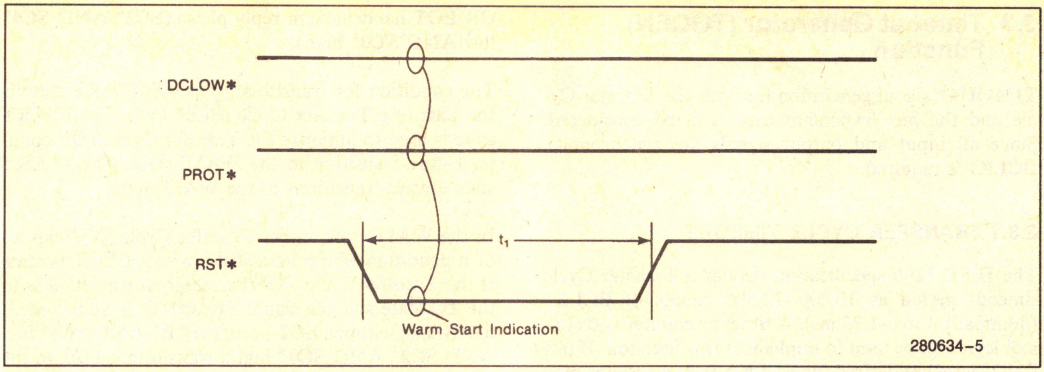


Figure 3-4. Warm Reset Timing on the PSB

Table 3-4. CSM Power Fail and Recovery Timing Specifications

Symbol	Parameter	Min	Max	Units
t_1	DC Power Hold from **DCLOW***	6.5		ms
t_2	**PROT*** Delay from **DCLOW***	6.0	6.25	ms
t_3	DC Power Setup to **DCLOW***	1.0		ms
t_4	**RST*** Delay from **DCLOW***	6.5	7.0	ms
t_5	**RST*** Setup from **DCLOW***	0.5		ms
t_6	**RST*** Active from **PROT***	50.0		ms
t_7	**DCLOW*** Pulse Width	7.5		ms
t_8	**PROT*** Hold from **DCLOW***	2.0	2.5	ms

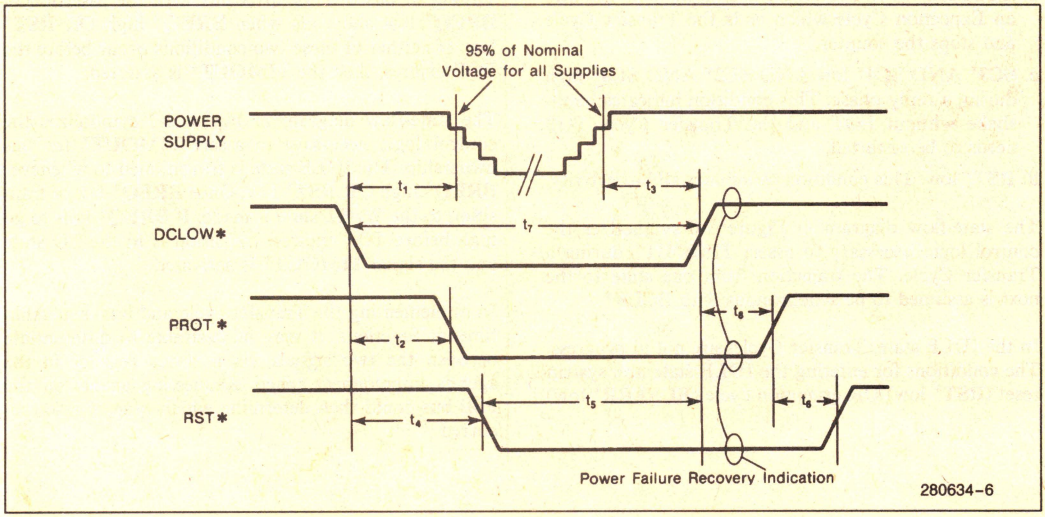


Figure 3-5. Power-Fail Recovery Timing on the PSB

3.3 Timeout Generator (TOGEN) Function

TIMOUT* signal generation for both the Transfer Cycle and the bus ownership cases will be considered. Since all input and output signals are synchronous, **BCLK*** is required.

3.3.1 TRANSFER CYCLE TIMEOUT

The IEEE 1296 specification defines a Transfer Cycle timeout period as 10,000–12,500 counts of **BCLK*** (nominally 1 ms–1.25 ms). A timer or counter and control logic can be used to implement this function. If test frequencies are desired for **BCLK***, then a counter may be a more desirable solution so the Transfer Cycle timeout period (TOP) will be a function of **BCLK*** and not fixed at 1 ms.

Timing of the Transfer Cycle begins on the first clock of a request phase; indicated by **SC0*** active. Once initiated and unless one of the following conditions is satisfied, the Transfer Cycle TOP will have expired and the CSM must assert **TIMOUT***:

a. **SC2*** AND **SC4*** low AND **SC0*** high during a reply phase. This condition indicates requester end of transfer (EOT) and replier ready handshake, which terminates Transfer Cycle timing.

b. **BUSERR*** low. This signal unconditionally initiates an Expection Cycle which ends the Transfer Cycle and stops the counter.

c. **SC3*** AND **SC4*** low AND **SC2*** AND **SC0*** high during a reply phase. This condition indicates handshake without EOT and the Transfer Cycle TOP needs to be restarted.

d. **RST*** low. This condition terminates all bus activity.

The state-flow diagram in Figure 3-6 symbolizes the control logic necessary to assert **TIMOUT*** during a Transfer Cycle. The transition from one state to the next is assumed to be synchronous with **BCLK***.

In the IDLE state, Transfer Cycles are not in progress. The conditions for entering the IDLE state are: system reset (**RST*** low) OR exception cycle (**BUSERR*** low) OR EOT handshake in reply phase (**SC2*** AND **SC4*** low AND **SC0*** high).

The condition for transitioning to the START state is the start of a Transfer Cycle (**SC0*** low). The START state is used to initialize the Transfer Cycle TOP counter before transition to the WAIT state. The START state always transitions to the WAIT state.

In the WAIT state, either Transfer Cycle TOP expires or a condition where handshake without EOT occurs. If the Transfer Cycle TOP has expired, transition is to the TO state and the signal **TIMOUT*** is activated. If handshake without EOT occurs (**SC3*** AND **SC4*** low AND **SC2*** AND **SC0*** high), transition is back to the START state to reinitialize the Transfer Cycle TOP counter and then returns to the WAIT state.

The TO state always transitions back to the IDLE state. Thus, in this design, **TIMOUT*** is asserted for one **BCLK***.

3.3.2 BUS OWNERSHIP TIMEOUT

The IEEE 1296 specification identifies the bus ownership timeout as system defined. A timer or counter and control logic can also be used to implement this function.

Timing of bus ownership begins with the assertion of **BREQ*** low and ends when **BREQ*** high OR **RST*** low. If neither of these two conditions occur before the TOP expires; then the **TIMOUT*** is asserted.

The state-flow diagram in Figure 3-7 symbolizes the control logic necessary to assert **TIMOUT*** for bus ownership. The IDLE state is transitioned to whenever **BREQ*** high OR **RST*** low. Once **BREQ*** low, a transition to the WAIT state is made. If **BREQ*** fails to go high before TOP expires, transition is to the TO state and the signal **TIMOUT*** is activated.

In implementing the Transfer cycle and bus ownership timeout functions, it may be desirable to differentiate between the two signals via a status register in the agent's interconnect space. Requesting agents on the PSB bus could then determine which type of reset occurred.

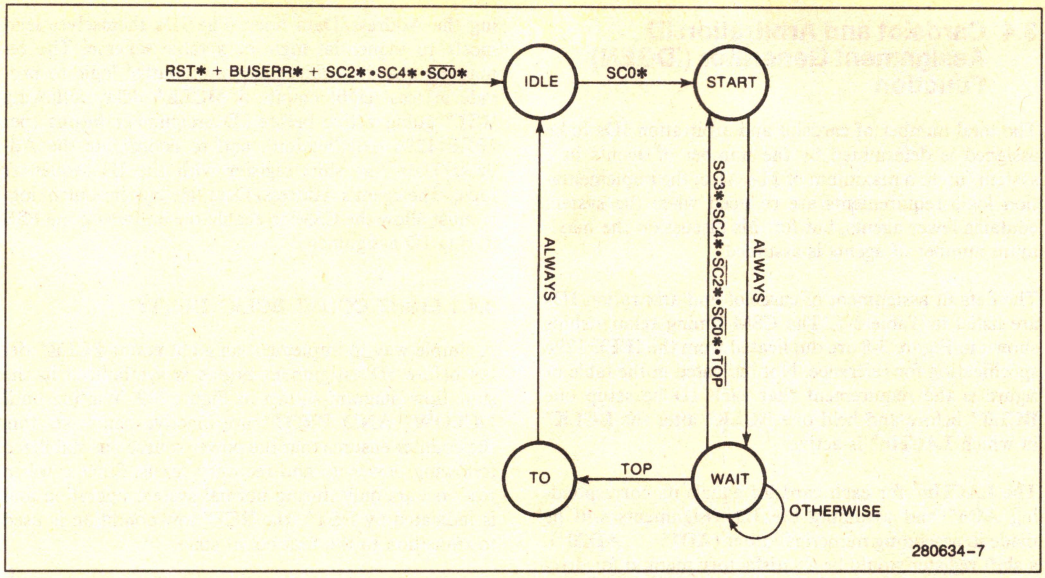


Figure 3-6. State-Flow Diagram for Monitoring Transfer Cycle Timeout

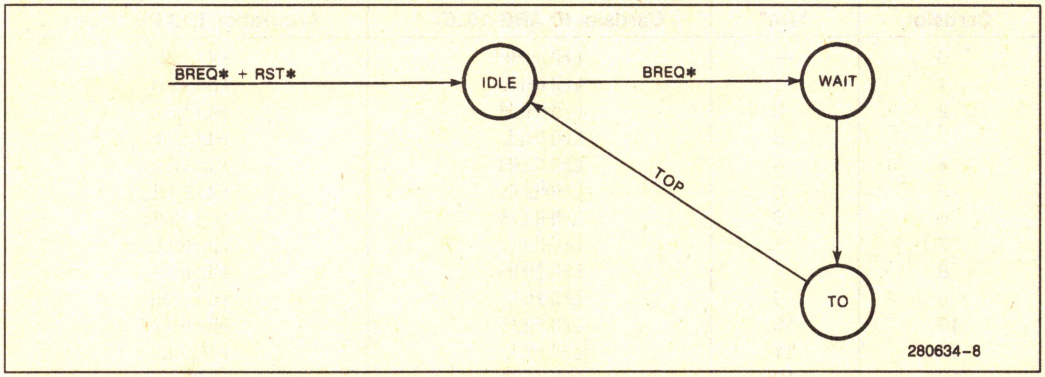


Figure 3-7. State-Flow Diagram for Monitoring Bus Ownership Timeout

3.4 Cardslot and Arbitration ID Assignment Generator (IDGEN) Function

The total number of cardslot and arbitration IDs to be assigned is determined by the number of agents in a system (up to a maximum of 21). Also, the implementation logic requirements are reduced when the system contains fewer agents, but for this discussion the maximum number of agents is assumed.

The default assignment of cardslot and arbitration IDs are listed in Table 3-5. The CSM timing relationships shown in Figure 3-8 are duplicated from the IEEE 1296 specification for reference. Not indicated in the table or figure is the requirement that each ID be setup one **BCLK*** before and held one **BCLK*** after the **BCLK*** in which **LACHn*** is active.

The **LACHn*** for each cardslot equals its corresponding **ADn*** and assuming the ID assignments will be made in ascending numerical order (**AD1*** . . . **AD20***), a shift register would be a satisfactory method for driving the Address/Data lines. The IDs themselves lend nicely to sequential logic or a table scheme. The remaining circuit requirements are control logic to provide at least eight counts of **BCLK*** delay following **RST*** going active before ID assignment begins (per IEEE 1296 specification), and to coordinate the Address/Data line shift register with the ID sequencer logic. The agent's Address/Data line buffer control logic must allow the CSM to enable the buffers on the PSB during ID assignment.

3.4.1 EIGHT COUNT BCLK* DELAY

A simple way to implement an eight count **BCLK*** delay before ID assignment begins is symbolized in the state-flow diagram shown in Figure 3-9. Waiting until **DCLOW*** AND **PROT*** are inactive before starting the counter ensures that the power source has stabilized following power-up and recovery resets. Since a warm reset occurs only during normal system operation and is indicated by **RST***, the **RST*** low condition is used to transition to the first count state.

Table 3-5. Default Cardslot and Arbitration ID Values

Cardslot	ADn*	Cardslot ID ARB<5..0>*	Arbitration ID ARB<5..0>*
0	—	LHHHHH	HLLLLL
1	1	LHHHHL	HLLLLH
2	2	LHHHLH	HLLLHL
3	3	LHHHLL	HLLLHH
4	4	LHHLHH	HLLHLL
5	5	LHHLHL	HLLHHL
6	6	LHHLLH	HLLHHH
7	7	LHHLLL	HLHLLL
8	8	LHLHHH	HLHHLL
9	9	LHLHHL	HLHHHL
10	10	LHLHLH	HLHHHH
11	11	LHLHLL	HHLLLL
12	12	LHLLHH	HHLLLH
13	13	LHLLHL	HHLLHH
14	14	LHLLLH	HHLHHH
15	15	LHLLLL	HHHLLL
16	16	LLHHHH	HHHLLH
17	17	LLHHHL	HHHLHH
18	18	LLHHLH	HHHHLL
19	19	LLHHLL	HHHHLH
20	20	LLHLHH	HHHHHL

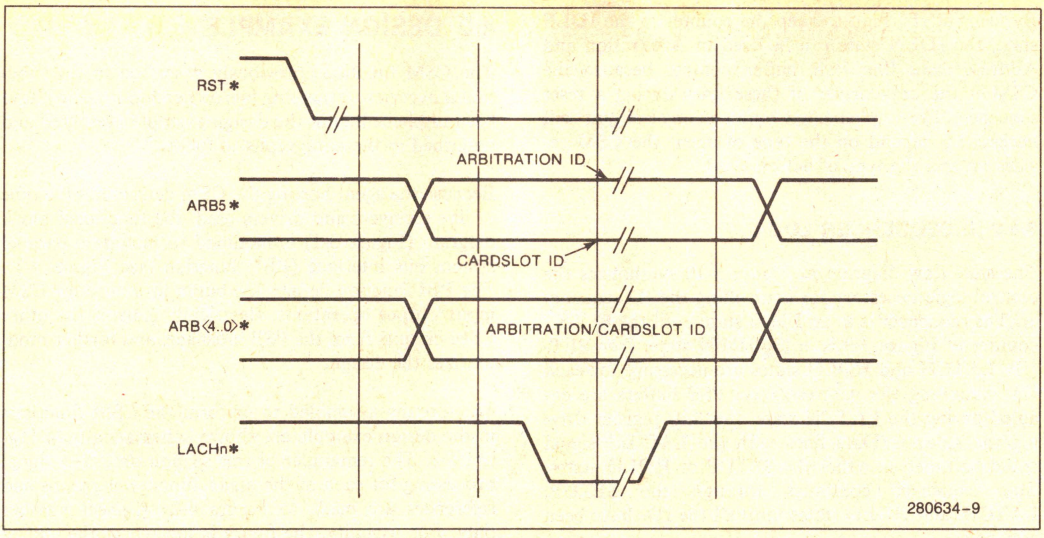


Figure 3-8. Cardslot and Arbitration ID Assignment Timing

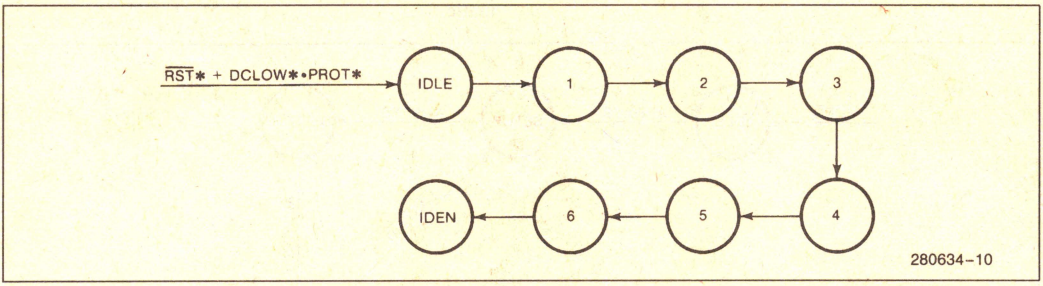


Figure 3-9. State-Flow Diagram for Delaying ID Assignment

5

By using **RST*** high to keep the counter in the IDLE state, the IDEN state can be used in Arbitration and Address/Data line PSB buffer control because the CSM is the only driver of these lines during a reset sequence. The implemented buffer control circuit will necessarily depend on the type of agent the CSM resides on and the type of buffers used.

3.4.2 ID SEQUENCER LOGIC

The state-flow diagram in Figure 3-10 symbolizes the control logic necessary for controlling the ID sequencer. The sequencer is in an IDLE state until the **BCLK*** counter of Figure 3-9 is in the IDEN state; then SETUP, LATCH and HOLD states are necessary for each ID. Assuming the Address/Data PSB buffers are enabled during the LATCH state, the shift register driving the Address/Data lines with the **LACHn*** signal could be timed in either the SETUP or HOLD states. The sequencer continues through the SETUP, LATCH, and HOLD states until all the IDs have been assigned.

4.0 DESIGN EXAMPLE

The CSM functions previously described in the functional overview (excepting bus ownership timeout) have been implemented in the design example presented and described in the paragraphs to follow.

Because the agent hosting the CSM determines the type of line receivers and drivers used, the functional block diagram (Figure 3-1) is modified to include a parallel system bus interface (PBI) function (see Figure 4-1). The PBI function defines the buffer structure for CSM input/output operations, electrically isolates the other CSM circuits from the PSB interface, and further modularizes the design.

The circuits assembled to perform the CSM functions in the design example are shown schematically in Figure 4-2. The remainder of this section describes signal processing for each of the major functional groups and references are made to Figure 4-2 by sheet number only. For usability, the figure is located at the end of this section following Figure 4-8. For simplicity, the decoupling capacities have been omitted from the schematic.

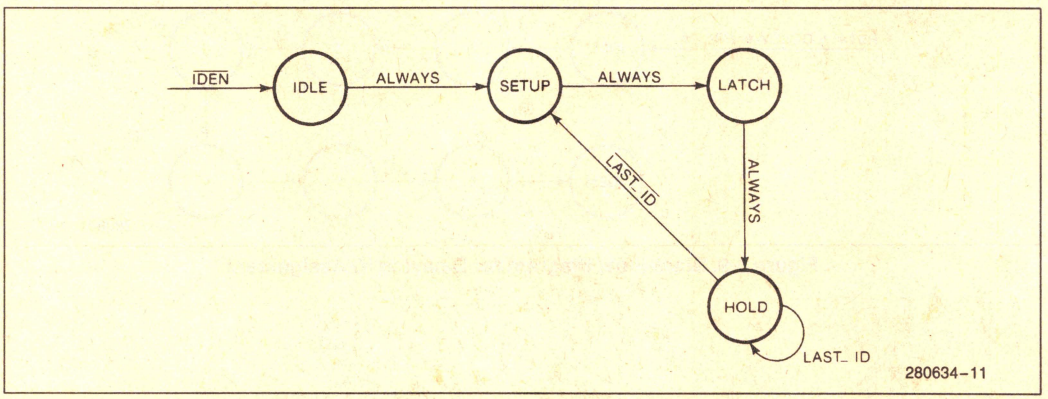


Figure 3-10. State-Flow Diagram for Controlling ID Assignment

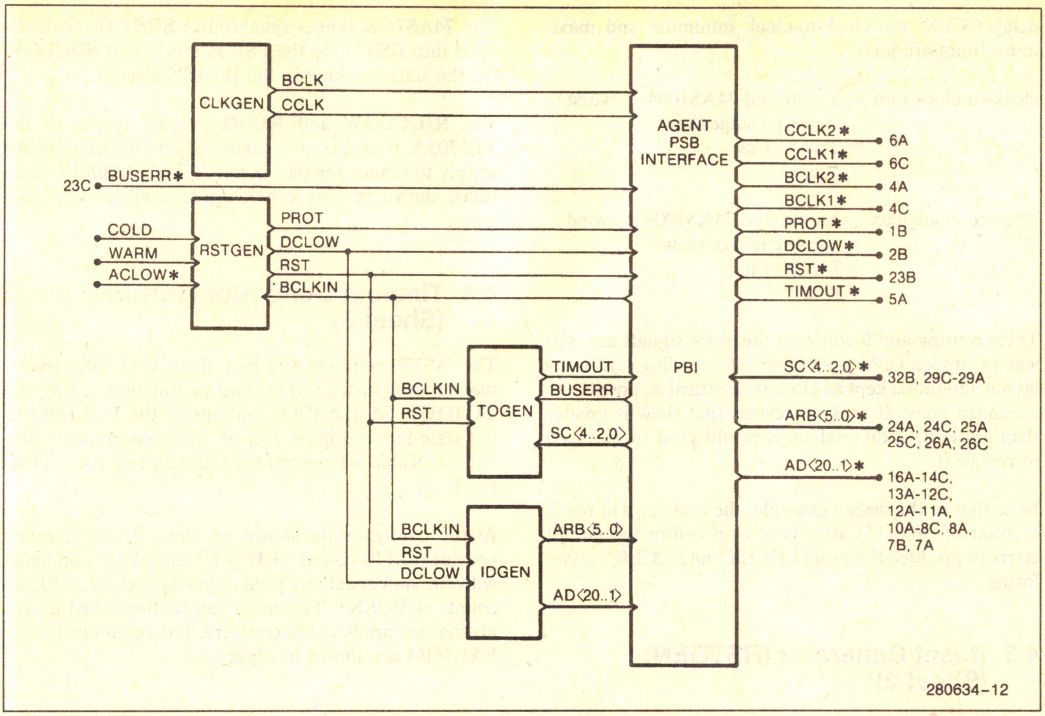


Figure 4-1. CSM Overall Block Diagram

4.1 PSB Buffer Interface (Sheet 1)

The majority of the PSB interface signals are buffered using 74F240 packages. The 74F240 circuitry meets the PSB loading and drive specifications and are satisfactory for this design example. The **ARB<5..0>*** lines are specified to operate open-collector and the 74S38 gates used meet this requirement. The **BUSERR*** input is buffered through a spare 74AS1004 gate simply to keep the part count down.

Since the CSM is the only driver of the clock, reset and timeout signals, their buffer enables are tied active. The **AD<20..1>*** lines are only driven by the CSM during reset sequences and the **ADEN*** signal from the IDGEN circuit is used to enable these lines. Also provided by the IDGEN circuit is the **IDEN** signal to enable the 74S38 gates during ID assignments. **BCLK1*** is buffered through a 74AS1004 gate as **BCLKIN** and provides the on-board timing for the synchronous logic.

4.2 Clock Generator (CLKGEN) (Sheet 2)

The 40 MHz oscillator, 74AS163 package and two 74AS1004 gates form the CLKGEN circuit. Dividing the 40 MHz to 20 MHz and 10 mHz in the same package controls clock-to-clock skew. Since all clock signals are driven by the same 74F240 package (sheet 1) and assuming a less than 0.5 ns same-package skew from both the 74AS163 and 74F240 packages, the worst case clock-to-clock is 1 ns (ignoring trace layout considertions). The IEEE 1296 specification defined clock-to-clock skew is listed in Table 3-1 as parameter t6. By using the 74AS1004 gates in the 20 MHz path to

delay **CCLK**, the clock-to-clock minimum and maximum times are met:

clock-to-clock min = 2 x mintpd 74AS1004 − worst case package skew
= 2 ns − 1 ns
= 1 ns

clock-to-clock max = 2 x maxtpd 74AS1004 + worst case package skew
= 8 ns + 1 ns
= 9 ns

Trace routing and loading on the clock signals are critical to proper CSM operation. The loading and trace layout should be kept as close to identical as possible to minimize skew. If analysis reveals that skew is greater than allowed, additional steps would need to be taken to reduce it.

Note that in the design example, the counter will reach a count of 0FH(15) after power-up before the circuit starts to produce the proper **BCLK*** and **CCLK*** waveforms.

4.3 Reset Generator (RSTGEN) (Sheet 2)

As described in paragraph 3.2, the RSTGEN function is influenced by the power supply used and system configuration. For simplicity, the design example assumes the following:

- No battery backup capability
- Equal ramp-up time on all power supply levels
- **ACLOW*** input from the power supply signalling eminent power failure
- Two active high debounced inputs for cold and warm reset invocation by the system user.

The TL7705A and 74AS74 packages on sheet 2 and part of the PAL16R4B programmable logic device on sheet 3 form the RSTGEN circuit. The timing for **DCLOW** and **ARST** are provided by the CT inputs on the TL7705A power supply monitors, which are adjustable to meet different power supply ramp-up times. The timing provided at the CT input can be determined by adding the power supply ramp-up time from the monitor threshold at 90 percent VCC to the minimum pulse width of **DCLOW** or **ARST**. (Note that the pulse width for **ARST** should also include the pulse width for **DCLOW**.) For this example, a 5 ms ramp-up from 4.5- to 4.5-Vdc was assumed.

The 74AS74 packages synchronize **SRST** (later developed into **RST***) for the PSB interface and **SDCLOW** for the state machine in the IDGEN circuit.

The **RIDCLOW** and **RIRST*** signal inputs to the TL7705A packages are generated in the PAL16R4B simply to reduce the part count. The PAL16R4B equations, shown in Figure 4-3, are basically that of OR gates.

4.4 Timeout Generator (TOGEN) (Sheet 3)

The 74S779 counters and part of the PAL16R4 implement the Transfer Cycle timeout function. Using the **TIMOUT, S1** and **CET*** outputs of the PAL16R4 as the state bits in Figure 3-6, the state assignments provided in Table 4-1 control the counters and assert **TIMOUT**.

In the configuration shown on sheet 3, the counters provide 10,240 counts of **BCLK*** and when combined with the state machine, yield a timeout period of 10,243 counts of **BCLK***. The count can be fine tuned by adjusting the inputs to the counters. The equations for the PAL16R4 are shown in Figure 4-3.

4.5 Cardslot and Arbitration ID Generator (IDGEN)

The 63RA481A PROM (ID sequencer add on sheet 3 and the 74LS164 AD* shift registers and PAL16R8 on sheet 4 form the IDGEN circuit. Note that to keep the part count down, the shift register for driving **AD<17..20>*** is implemented in the PAL16R8. Also implemented in the PAL16R8 is the IDEN state machine depicted in Figure 3-9. The implemented IDEN state machine only provides one **BCLK*** delay instead of eight, but the ID sequencer provides the additional counts required by sequencing through unused cardslot IDs. (The signal to **IDEN2** is identical to **IDEN** and is used to furnish additional D.C. drive for the ID sequencer.)

Table 4-1. TOGEN State Assignments

State	TIMOUT	S1	CET*
IDLE	0	0	1
START	0	0	0
WAIT	0	1	0
TO	1	1	1

```
chip name PAL16R4

BCLKIN SRST SC0 SC2 SC3 SC4 BUSERR /ACLOW /TOP GND
/OE /RIRST /RIDCLOW S1 /CET TIMOUT RST COLD WARM VCC

equations

/TIMOUT     :=  /TOP * /TIMOUT * CET
            + /TIMOUT * /S1
            + TIMOUT * S1 * /CET
            + /SC0 * SC2 * SC4
            + BUSERR
            + RST

/S1         :=  TIMOUT * S1 * /CET
            + /SC0 * SC2 * SC4
            + BUSERR
            + RST
            + /SC0 * SC3 * SC4 * /TOP * /TIMOUT * S1 * CET
            + /TIMOUT * /S1 * /CET

CET         :=  /RST * /SC2 * /BUSERR * /TOP * /TIMOUT * CET
            + /RST * SC0 * /BUSERR * /TIMOUT * /S1
            + /RST * /SC4 * /BUSERR * /TOP * /TIMOUT * CET
            + /RST * SC0 * /BUSERR * /TOP * /TIMOUT * CET
            + /RST * /SC4 * /BUSERR * /TIMOUT * /S1 * CET
            + /RST * /SC2 * /BUSERR * /TIMOUT * /S1 * CET

RIDCLOW :=  COLD
         + ACLOW

RIRST    :=  COLD
         + WARM

/RST     :=  /SRST
```

280634-13

Figure 4-3. Equations for TOGEN PAL16R4B

Six outputs of the ID sequencer are used to drive the **ARB<5..0>*** lines and the remaining two outputs are assigned the signal names **LATCH** and **HOLD**. The IDLE state, shown in Figure 3-10, as indicated when all of the ID sequencer outputs are low; the SETUP state by the **ARB<5..0>*** outputs changing to a new ID; the LATCH state when the **LATCH** output is active high and the **HOLD** output is inactive low; the HOLD state when the **HOLD** output is active high and the **LATCH** output inactive low. The ID sequencer remains in the last HOLD state until **IDEN** becomes inactive low and is then reset to the IDLE state.

The state-flow diagrams in Figures 3-9 and 3-10 are modified as shown in Figures 4-4 and 4-5. These modifications take advantage of the design implemented to supply the eight counts of **BCLK*** delay before assigning IDs and to initialize the 74LS164 AD* shift registers.

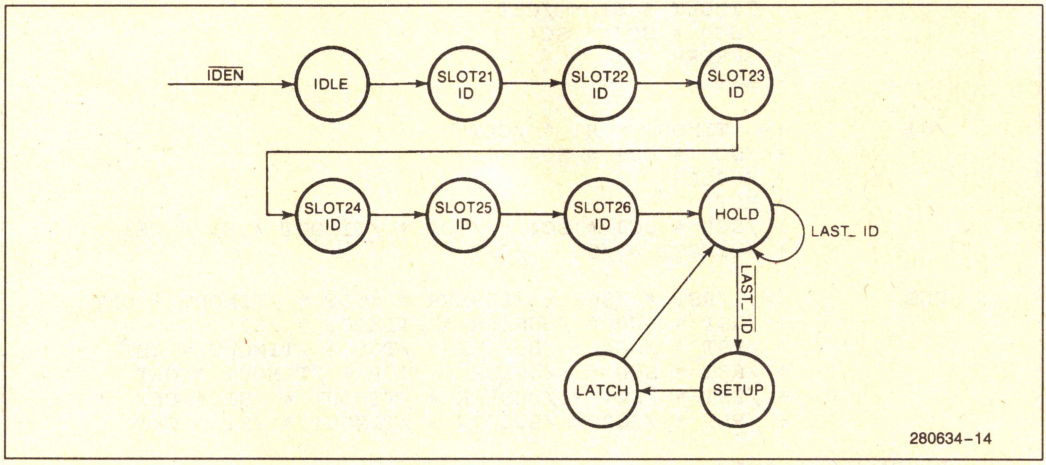


Figure 4-4. Modified State-Flow Diagrams for ID Assignment

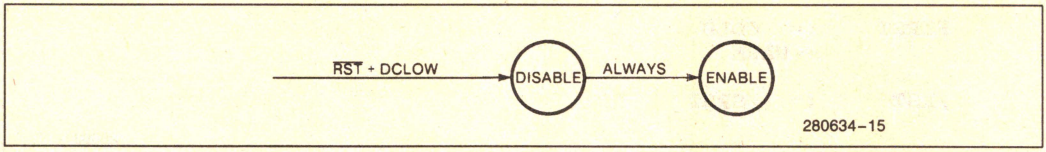


Figure 4-5. Modified IDEN State-Flow Diagram

The AD* shift registers are clocked during the HOLD state of the ID sequencer. Two passes are needed through these registers, one to latch cardslot IDs and one to latch arbitration IDs. The AD* shift registers are initialized at zero while **IDEN** is inactive low. During the HOLD state of unused cardslot ID 26 and the HOLD state of cardslot ID 20, the **SDATA** (Serial Data) output of the PAL16R8A is active high so that a one is shifted into the AD* shift registers at the beginning of each pass.

The **LATCH** output from the ID sequencer is inverted to produce **ADEN***. This signal enables the **AD<20..1>*** buffers (sheet 1) onto the PSB bus during the **LATCH** state.

Figure 4-6 shows the timing produced by the IDGEN circuit. The equations for the PAL16R8 are shown in Figure 4-7. The PROM ID code information is provided in Figure 4-8.

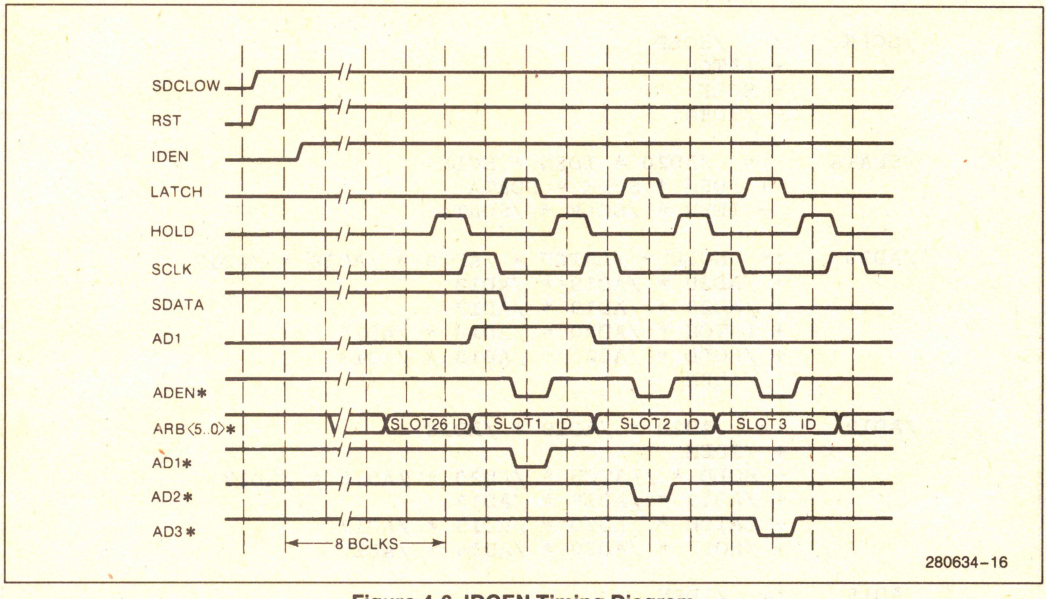


Figure 4-6. IDGEN Timing Diagram

```
chip name PAL16R8
BCLKIN RST SDCLOW HOLD LATCH AD16 IC7 IC8 IC9 GND
/OE AD17 AD18 AD19 AD20 SCLK SDATA IDEN2 IDEN VCC
equations
/IDEN     :=  SDCLOW
          + /RST

/IDEN2    :=  SDCLOW
          + /RST

/SCLK     :=  /HOLD
          + LATCH
          + SCLK
          + /IDEN

/SDATA     :=  /AD20 * IDEN * SCLK
           + IDEN * SCLK * SDATA
           + IDEN * /SCLK * /SDATA

/AD20     :=  HOLD * /LATCH * /AD19 * /AD18 * /AD17
          + /AD20 * /AD19 * /AD18
          + /AD20 * /AD19 * /AD17
          + LATCH * /AD20 * /AD18 * /AD17
          + /HOLD * /AD20 * /AD18 * /AD17
          + /IDEN

/AD19     :=  /AD20 * /AD19 * /AD18
          + /IDEN
          + HOLD * /LATCH * /AD20 * /AD18 * /AD17
          + /AD19 * /AD18 * /AD17
          + LATCH * /AD20 * /AD19 * /AD17
          + /HOLD * /AD20 * /AD19 * /AD17

/AD18     :=  /IDEN
          + /AD19 * /AD18 * /AD17
          + HOLD * /LATCH * /AD20 * /AD19 * /AD17
          + /AD20 * /AD18 * /AD17
          + LATCH * /AD20 * /AD19 * /AD18
          + /HOLD * /AD20 * /AD19 * /AD18

/AD17     :=  /IDEN
          + LATCH * /AD20 * /AD19 * /AD17
          + /HOLD * /AD20 * /AD19 * /AD17
          + /AD16 * /AD20 * /AD19 * /AD17
          + HOLD * /LATCH * /AD20 * /AD19 * /AD18 * AD17
          + /AD20 * /AD19 * AD18 * /AD17
          + /AD20 * AD19 * /AD18 * /AD17
          + AD20 * /AD19 * /AD18 * /AD17
```

280634-17

Figure 4-7. Equations for IDGEN PAL16R8A

```
ADDRESS     DATA        COMMENTS
00000000    00110101    ;initial state
00110101    00110110    ;setup state for cardslot id 21
00110110    00110111    ;setup state for cardslot id 22
00110111    00111000    ;setup state for cardslot id 23
00111000    00111001    ;setup state for cardslot id 24
00111001    00111010    ;setup state for cardslot id 25
00111010    10111010    ;setup state for cardslot id 26
10111010    00100001    ;hold state for cardslot id 26
00100001    01100001    ;setup state for cardslot id 1
01100001    10100001    ;latch state for cardslot id 1
10100001    00100010    ;hold state for cardslot id 1
00100010    01100010    ;setup state for cardslot id 2
01100010    10100010    ;latch state for cardslot id 2
10100010    00100011    ;hold state for cardslot id 2
00100011    01100011    ;setup state for cardslot id 3
01100011    10100011    ;latch state for cardslot id 3
10100011    00100100    ;hold state for cardslot id 3
00100100    01100100    ;setup state for cardslot id 4
01100100    10100100    ;latch state for cardslot id 4
10100100    00100101    ;hold state for cardslot id 4
00100101    01100101    ;setup state for cardslot id 5
01100101    10100101    ;latch state for cardslot id 5
10100101    00100110    ;hold state for cardslot id 5
00100110    01100110    ;setup state for cardslot id 6
01100110    10100110    ;latch state for cardslot id 6
10100110    00100111    ;hold state for cardslot id 6
00100111    01100111    ;setup state for cardslot id 7
01100111    10100111    ;latch state for cardslot id 7
10100111    00101000    ;hold state for cardslot id 7
00101000    01101000    ;setup state for cardslot id 8
01101000    10101000    ;latch state for cardslot id 8
10101000    00101001    ;hold state for cardslot id 8
00101001    01101001    ;setup state for cardslot id 9
01101001    10101001    ;latch state for cardslot id 9
10101001    00101010    ;hold state for cardslot id 9
00101010    01101010    ;setup state for cardslot id 10
01101010    10101010    ;latch state for cardslot id 10
10101010    00101011    ;hold state for cardslot id 10
00101011    01101011    ;setup state for cardslot id 11
01101011    10101011    ;latch state for cardslot id 11
10101011    00101100    ;hold state for cardslot id 11
00101100    01101100    ;setup state for cardslot id 12
01101100    10101100    ;latch state for cardslot id 12
```

280634-18

Figure 4-8. IDGEN PROM Content (Sheet 1 of 3)

```
ADDRESS     DATA        COMMENTS
10101100    00101101    ;hold state for cardslot id 12
00101101    01101101    ;setup state for cardslot id 13
01101101    10101101    ;latch state for cardslot id 13
10101101    00101110    ;hold state for cardslot id 13
00101110    01101110    ;setup state for cardslot id 14
01101110    10101110    ;latch state for cardslot id 14
10101110    00101111    ;hold state for cardslot id 14
00101111    01101111    ;setup state for cardslot id 15
01101111    10101111    ;latch state for cardslot id 15
10101111    00110000    ;hold state for cardslot id 15
00110000    01110000    ;setup state for cardslot id 16
01110000    10110000    ;latch state for cardslot id 16
10110000    00110001    ;hold state for cardslot id 16
00110001    01110001    ;setup state for cardslot id 17
01110001    10110001    ;latch state for cardslot id 17
10110001    00110010    ;hold state for cardslot id 17
00110010    01110010    ;setup state for cardslot id 18
01110010    10110010    ;latch state for cardslot id 18
10110010    00110011    ;hold state for cardslot id 18
00110011    01110011    ;setup state for cardslot id 19
01110011    10110011    ;latch state for cardslot id 19
10110011    00110100    ;hold state for cardslot id 19
00110100    01110100    ;setup state for cardslot id 20
01110100    10110100    ;latch state for cardslot id 20
10110100    00011110    ;hold state for cardslot id 20
00011110    01011110    ;setup state for arbitration id 1
01011110    10011110    ;latch state for arbitration id 1
10011110    00011101    ;hold state for arbitration id 1
00011101    01011101    ;setup state for arbitration id 2
01011101    10011101    ;latch state for arbitration id 2
10011101    00011100    ;hold state for arbitration id 2
00011100    01011100    ;setup state for arbitration id 3
01011100    10011100    ;latch state for arbitration id 3
10011100    00011011    ;hold state for arbitration id 3
00011011    01011011    ;setup state for arbitration id 4
01011011    10011011    ;latch state for arbitration id 4
10011011    00011001    ;hold state for arbitration id 4
00011001    01011001    ;setup state for arbitration id 5
01011001    10011001    ;latch state for arbitration id 5
10011001    00011000    ;hold state for arbitration id 5
00011000    01011000    ;setup state for arbitration id 6
01011000    10011000    ;latch state for arbitration id 6
10011000    00010111    ;hold state for arbitration id 6
00010111    01010111    ;setup state for arbitration id 7
```

280634-19

Figure 4-8. IDGEN PROM Content (Sheet 2 of 3) (Continued)

```
ADDRESS     DATE        COMMENTS
01010111    10010111    ;latch state for arbitration id 7
10010111    00010011    ;hold state for arbitration id 7
00010011    01010011    ;setup state for arbitration id 8
01010011    10010011    ;latch state for arbitration id 8
10010011    00010001    ;hold state for arbitration id 8
00010001    01010001    ;setup state for arbitration id 9
01010001    10010001    ;latch state for arbitration id 9
10010001    00010000    ;hold state for arbitration id 9
00010000    01010000    ;setup state for arbitration id 10
01010000    10010000    ;latch state for arbitration id 10
10010000    00001111    ;hold state for arbitration id 10
00001111    01001111    ;setup state for arbitration id 11
01001111    10001111    ;latch state for arbitration id 11
10001111    00001110    ;hold state for arbitration id 11
00001110    01001110    ;setup state for arbitration id 12
01001110    10001110    ;latch state for arbitration id 12
10001110    00001100    ;hold state for arbitration id 12
00001100    01001100    ;setup state for arbitration id 13
01001100    10001100    ;latch state for arbitration id 13
10001100    00001000    ;hold state for arbitration id 13
00001000    01001000    ;setup state for arbitration id 14
01001000    10001000    ;latch state for arbitration id 14
10001000    00000111    ;hold state for arbitration id 14
00000111    01000111    ;setup state for arbitration id 15
01000111    10000111    ;latch state for arbitration id 15
10000111    00000110    ;hold state for arbitration id 15
00000110    01000110    ;setup state for arbitration id 16
01000110    10000110    ;latch state for arbitration id 16
10000110    00000100    ;hold state for arbitration id 16
00000100    01000100    ;setup state for arbitration id 17
01000100    10000100    ;latch state for arbitration id 17
10000100    00000011    ;hold state for arbitration id 17
00000011    01000011    ;setup state for arbitration id 18
01000011    10000011    ;latch state for arbitration id 18
10000011    00000010    ;hold state for arbitration id 18
00000010    01000010    ;setup state for arbitration id 19
01000010    10000010    ;latch state for arbitration id 19
10000010    00000001    ;hold state for arbitration id 19
00000001    01000001    ;setup state for arbitration id 20
01000001    10000001    ;latch state for arbitration id 20
10000001    10000001    ;hold state for arbitration id 20
end
```

280634-20

Figure 4-8. IDGEN PROM Content (Sheet 3 of 3) (Continued)

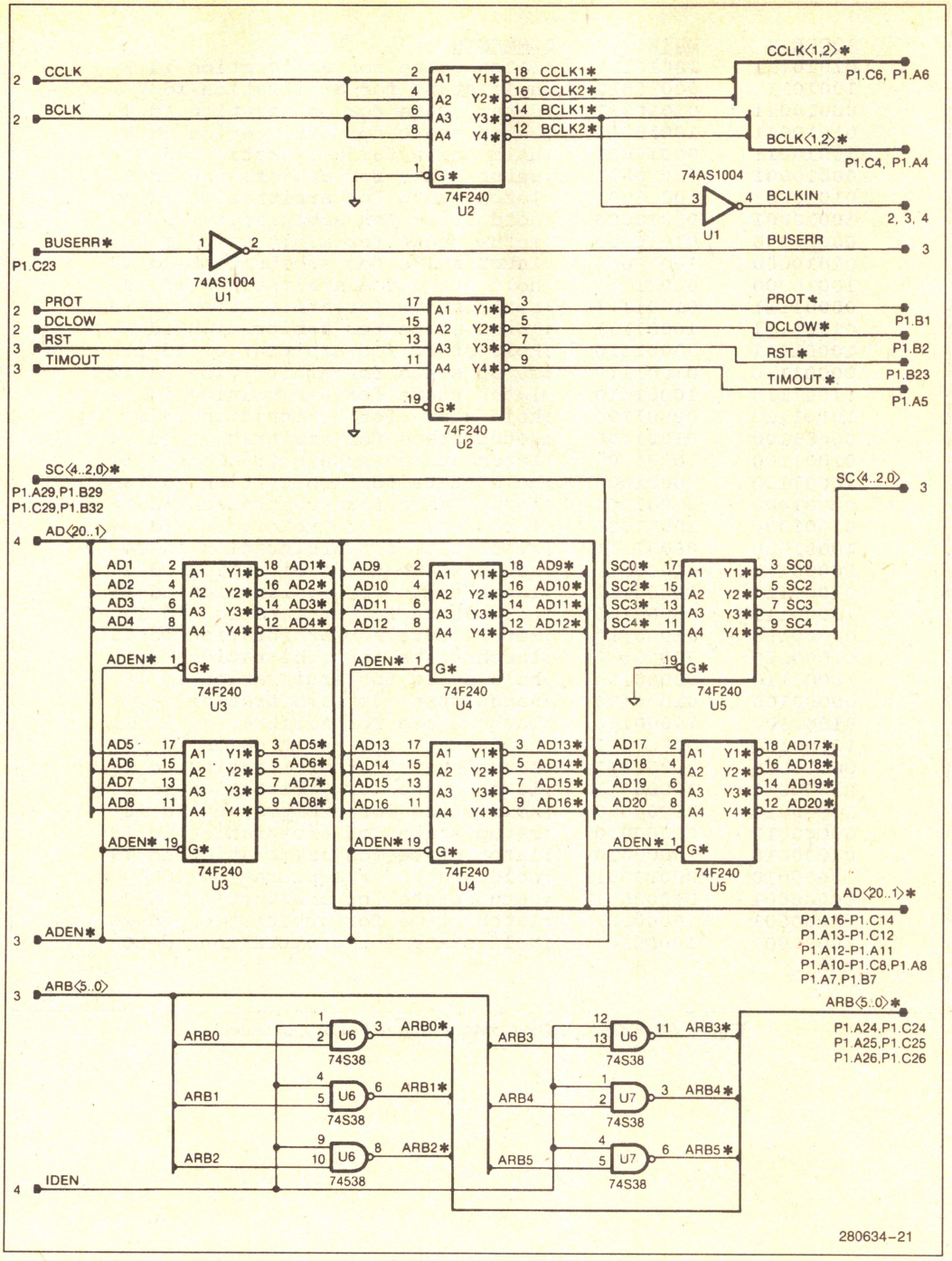


Figure 4-9. CSM Functional Schematic Diagram (Sheet 1 of 4)

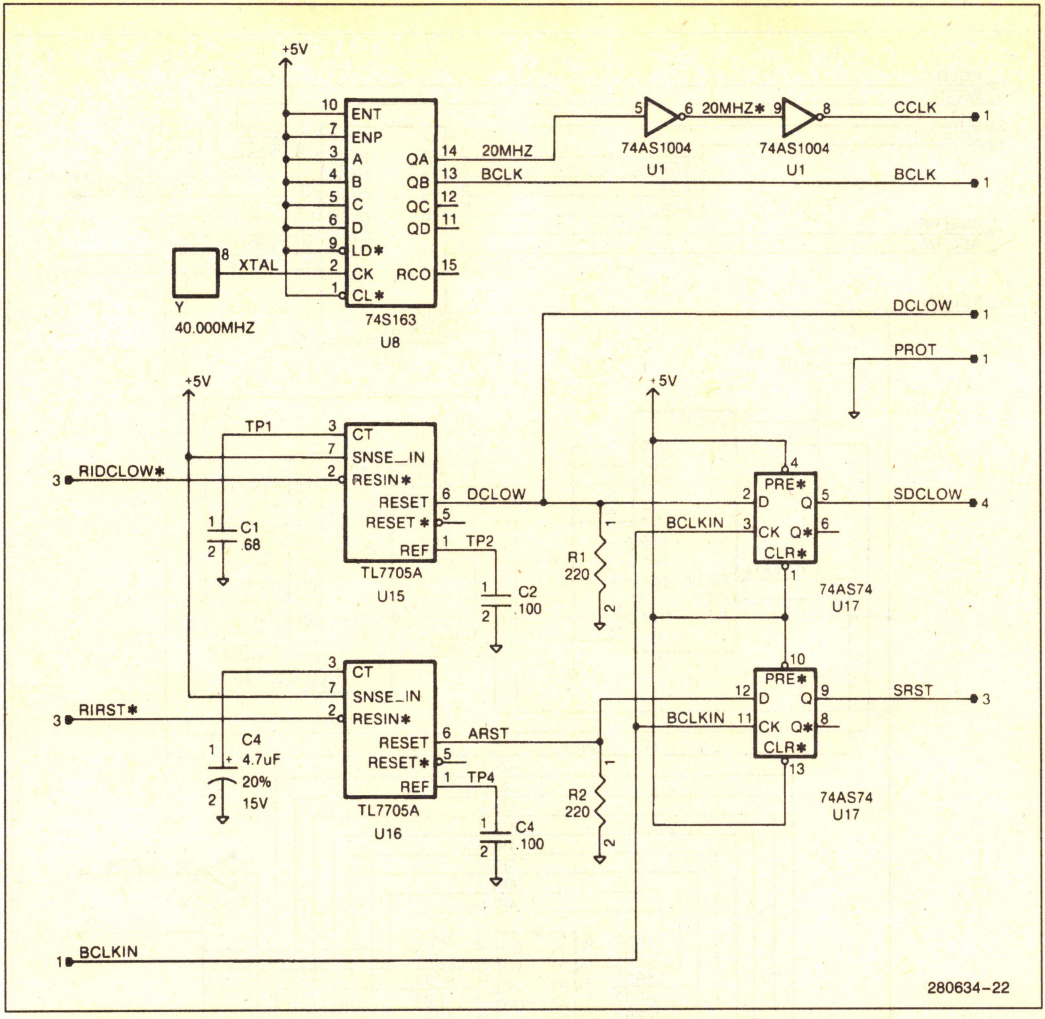


Figure 4-9. CSM Functional Schematic Diagram (Sheet 2 of 4) (Continued)

5

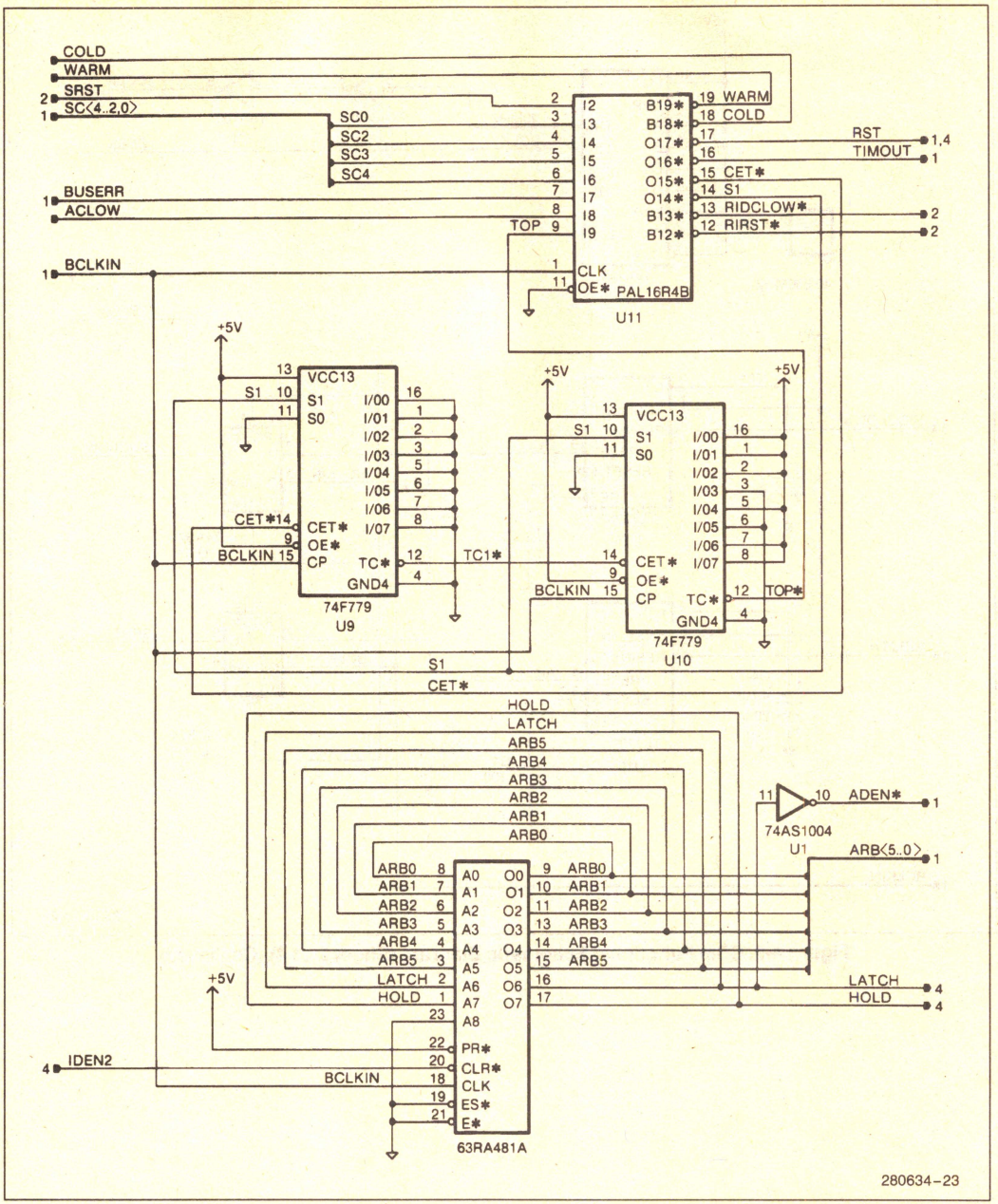


Figure 4-9. CSM Functional Schematic Diagram (Sheet 3 of 4) (Continued)

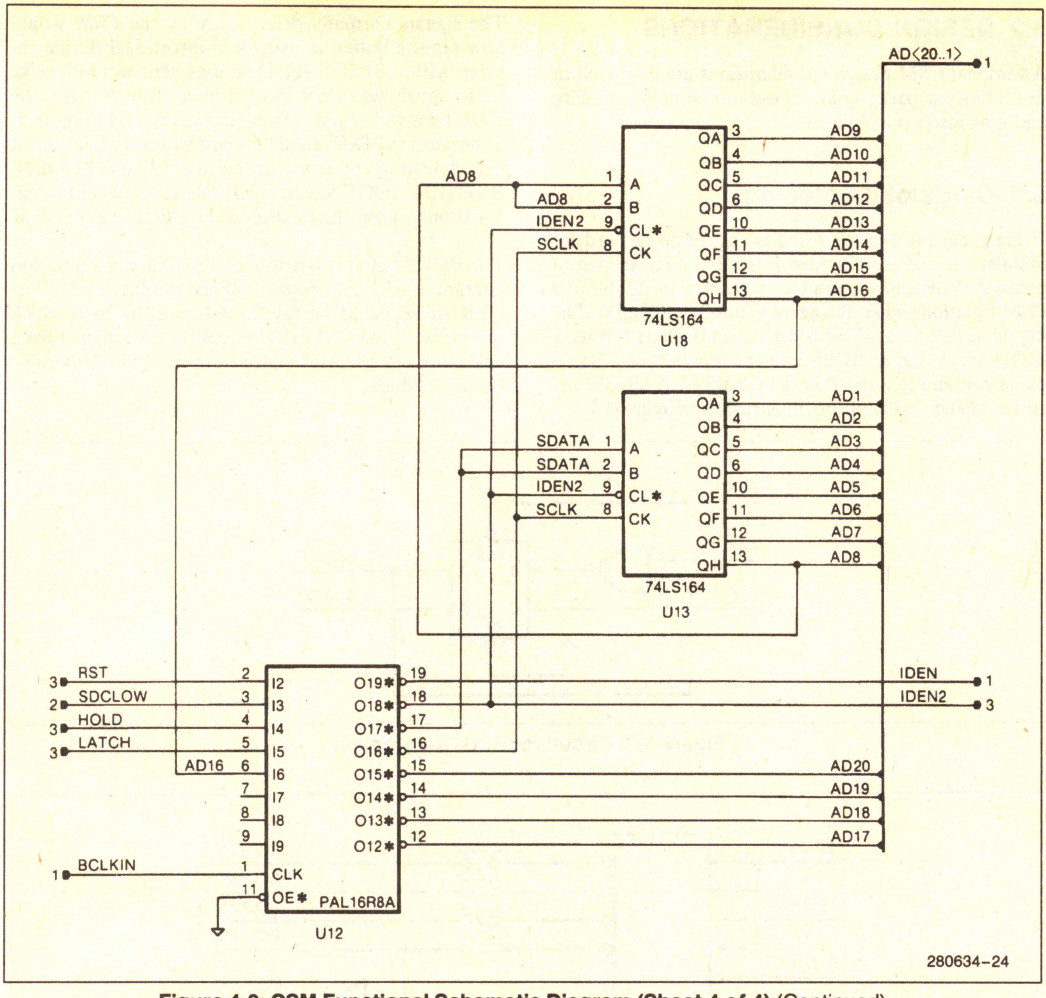


Figure 4-9. CSM Functional Schematic Diagram (Sheet 4 of 4) (Continued)

5.0 DESIGN CONSIDERATIONS

Additional CSM design considerations are discussed in the following paragraphs. These are cardslot mobility and diagnostic possibilities.

5.1 Cardslot 0 Detection

If the agent hosting the CSM is desired to be cardslot mobile, i.e. operate the cardslots other than 0; then a cardslot 0 detection circuit is necessary to disable the CSM functions when the agent is not in cardslot 0. The 4A pin is 0-Vdc in all cardslots except 0 where it is used to bus clock signal **BCLK*** to the left half of a backplane containing more than 12 cardslots. A simple circuit to detect cardslot 0 is illustrated in Figure 5-1.

The signals normally driven only by the CSM would now require buffering using a bi-directional device enabled with a **SLOT0** signal so the agent would receive these signals when not located in cardslot 0. Also, the clock lines will require jumpering as shown in Figure 5-2, because the D.C. signal specifications do not permit driver loading of a signal receiver. The CLKGEN, RSTGEN, TOGEN, and IDGEN circuits would be enabled only when the cardslot 0 detection signal is true.

The **POR*** signal is a power-on reset having a duration guaranteeing stable power supply output levels. The **CLK** signal could be the 20 MHz or 10 MHz clock output from the CLKGEN circuit or be supplied from any other clock operational before the PSB clock drivers are enabled.

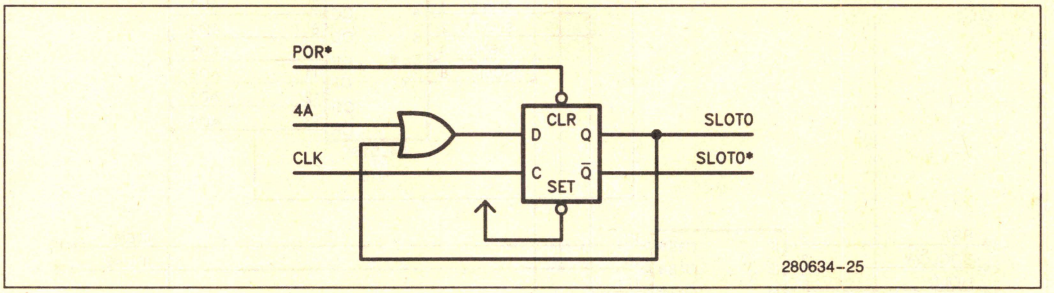


Figure 5-1. Circuit for SLOT0 Detection

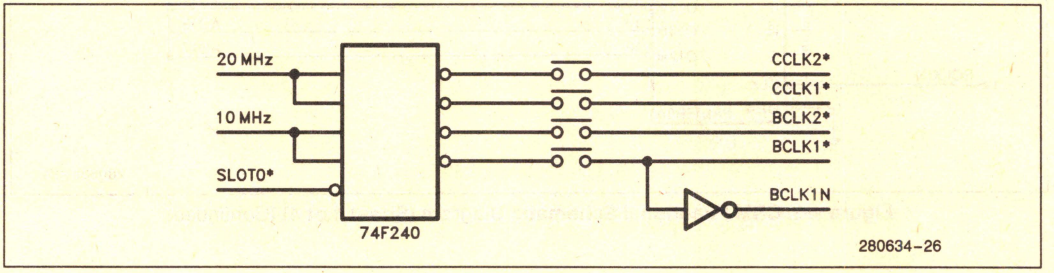


Figure 5-2. CSM Clock Jumper

5.2 System Diagnostic Options

For system test and diagnostic purposes, it may be desirable to provide for the capability of selecting other signal frequencies to exercise the clock drivers. For example, outputs from the frequency divider used in the design example could be selectively jumpered as inputs to the **BCLK*** and **CCLK*** line drivers.

A bus timeout disable function may also be desirable. Such a function can be implemented by providing an additional input to the TOGEN state machine driven by either the interconnect controller or jumper selectable.

5.3 CSM Functions on the Backplane

By removing the PBI function and interfacing directly to an agent's PSB interface, the design example in paragraph 4 could be added to almost any type of MULTIBUS II agent. As mentioned in paragraph 3.0, this agent might be a CPU board or a simple I/O replier device, which may contain other centralized system services.

The IEEE 1296 specification does not preclude putting slot 0 on the reverse side of the backplane. The minimal functionality described here will fit onto a small printed circuit card mounted on the reverse of the backplane—this does, of course require a backplane designed for this application but if you are trying to squeeze an "extra" slot into a 19 inch rack, this can be accomplished.

intel®

AP-423

APPLICATION NOTE

The Multibus II Interconnect Design Guide

JORY RADKE
MODULES DEVELOPMENT ENGINEER

ROGER FINGER
MULTIBUS II TECHNICAL MARKETING MANAGER

October 1990

Order Number: 280640-001

The Multibus II Interconnect Design Guide

CONTENTS

PAGE

1.0 INTERCONNECT ADDRESS SPACE ON MULTIBUS II 5-50

1.1 Who Should Read This Document? 5-50

1.2 Configuration Issues in Microcomputer Based Systems 5-50

1.3 Overview of Interconnect Address Space 5-50

1.3.1 Geographical Addressing 5-50

1.3.2 Microcontrollers in the Bus Interface 5-51

1.3.3 Addressing of Interconnect .. 5-52

1.3.4 Data Structures in Interconnect 5-52

1.3.5 Access Rights and Protection Records 5-53

1.3.6 Diagnostic Philosophy of Multibus II 5-54

1.4 Interconnect—the Manufacturer's Perspective 5-54

1.5 Assumptions Regarding Your Operating Environment 5-54

1.6 Design Methodology—An Overview 5-55

2.0 PREPARATION FOR USE 5-55

2.1 Hardware and Software Requirements 5-55

2.2 Loading Software onto Your System 5-56

2.3 Interconnect the Easy Way 5-56

3.0 THEORY OF OPERATION 5-57

3.1 MPC To Microcontroller Hardware Interface 5-57

3.2 MPC Interconnect Bus Registers 5-57

CONTENTS

PAGE

3.3 Partitioning of Functions between the Microcontroller, CPU, and MPC 5-58

3.3.1 MPC Diagnostic Testing 5-58

3.3.2 Dual-Port Memory Control ... 5-59

3.3.3 Message Retry Operations ... 5-59

3.3.4 Fail-Safe Counter Functions 5-59

3.4 Multibus II Reset Operations 5-59

3.4.1 Cold-Start and Recovery Reset 5-60

3.4.2 Warm-Start Reset 5-60

3.4.3 Local Reset 5-61

3.5 Initialization of the Microcontroller following Reset 5-61

3.6 Interrupt Driven Firmware Operations 5-61

3.6.1 Interconnect Interrupt Handler 5-62

3.6.2 Interconnect Read Operations 5-62

3.6.3 Interconnect Write Operations 5-64

3.6.4 Reset Interrupt Handler 5-64

3.6.5 Timer Based Functions 5-64

3.7 Use of Multibus II Boards with In-Circuit Emulators 5-64

3.8 Avoiding iPSB Timeouts 5-65

4.0 INTERCONNECT HARDWARE DESIGN 5-65

4.1 Microcontroller Pin Assignments .. 5-65

4.1.1 The Interconnect Address/ Data Bus 5-65

4.1.2 Reserved Pins on the 8751 .. 5-65

CONTENTS

PAGE

4.2 External TTL Options 5-65
4.2.1 Microcontroller Input Options 5-66
4.2.2 Microcontroller Output Options 5-67
4.2.3 LED Outputs 5-68
4.2.4 Output Enables and Chip Selects 5-68
4.2.5 Address Decode on the IAD Bus 5-68
4.2.6 Comparators 5-69
4.3 Bidirectional I/O on the IAD Bus .. 5-69

5.0 DEFINING THE INTERCONNECT TEMPLATE 5-70
5.1 What Goes into Interconnect 5-70
5.1.1 Core Functions Supplied 5-70
5.1.2 Getting Started: The Interconnect Worksheet 5-70
5.2 The Interconnect Header Record 5-70
5.2.1 Vendor ID Registers (0–1) ... 5-70
5.2.2 Board ID Registers (2–11) ... 5-72
5.2.3 Hardware Test Revision Number Register (16) 5-72
5.2.4 Class ID Register (17) 5-72
5.2.5 BIST Support Level Register (26) 5-72
5.3 Optional User Functions 5-72
5.4 Completing the Interconnect Configuration Worksheet 5-72
5.4.1 Global Access Rights 5-73
5.4.2 RAM Usage 5-73
5.4.3 Default Value 5-73
5.4.4 Read Routine Entry 5-73
5.4.5 Edit Routine Table 5-74
5.4.6 Write Routine Number 5-75
5.4.7 Constants for Value Checking 5-75

CONTENTS

PAGE

5.4.8 The Edit Entry Table 5-75
5.4.9 Symbolic Reference to RAM Locations 5-75
5.5 Loading the Macro Tables 5-75
5.5.1 The ENTAB Table 5-75
5.5.2 The EETGEN Table 5-75
5.5.3 The EXTERNS Macro: Generating External Label 5-76
5.6 Programming Techniques for Read, Write, and Edit Routines 5-76
5.6.1 Retrieving the Constant List from the EETABLE 5-76
5.6.2 Handling of Value Errors 5-77
5.7 Hooks for User Supplied Routines 5-77
5.7.1 INIT User: Custom Initialization Code 5-77
5.7.2 RST User: Special Handling upon Warm Reset 5-77
5.7.3 Poll User: Polled User Functions 5-77
5.7.4 User Timer: Timer Based Functions 5-77
5.8 Generating the Object Module 5-78
5.8.1 Checking Microcontroller Resource Utilization 5-78
5.8.2 Checking Critical Timing Paths 5-79
5.8.3 Programming the 8751 5-81

6.0 FUNCTIONAL TESTING OF THE INTERCONNECT SUBSYSTEM 5-81
6.1 Testing the Interconnect Subsystem 5-81
6.2 Power-On and Initialization Functions 5-81
6.3 The Interconnect Configuration Utility 5-81
6.4 A Test Methodology for Interconnect Functions 5-82
6.5 Summary and Conclusion 5-83

CONTENTS

PAGE

APPENDIX A BIBLIOGRAPHY OF RELATED READING 5-87

APPENDIX B FLOWCHARTS OF THE CORE IC FIRMWARE 5-88

CONTENTS

PAGE

APPENDIX C READ, WRITE, AND EDIT ROUTINES 5-93

APPENDIX D USER CALLABLE MACRO ROUTINES 5-119

List of Figures

Figure 1-1. Interconnect Overview 5-51
Figure 1-2. Hardware Support for Message Passing 5-51
Figure 1-3. Interconnect Addressing 5-52
Figure 1-4. Function Records in Interconnect Core Firmware 5-53
Figure 1-5. Firmware Content of Multibus II Boards 5-54
Figure 3-1. MPC to Interconnect Pathway 5-57
Figure 3-2. MPC Interconnect Registers 5-58
Figure 3-3. MPC Fail-Safe Counters 5-59
Figure 3-4. Reset Hardware 5-60
Figure 3-5. Microcontroller Interrupt Sources 5-62
Figure 3-6. Indirect Jump Tables for Read Routines 5-62
Figure 3-7. Indirect Jump Tables for Edit and Write Routines 5-63
Figure 4-1. Interconnect Core Hardware Design 5-66
Figure 4-2. Microcontroller Input Options 5-67
Figure 4-3. Microcontroller Output Options 5-68
Figure 4-4. Bidirectional I/O on the IAD Bus 5-69
Figure 5-1. Interconnect Configuration Worksheet (2 sheets) 5-71
Figure 5-2. ENTAB Macro Source Code Listing (2 sheets) 5-74
Figure 5-3. EEGEN Macro Source Code Listing 5-76
Figure 5-4. Microcontroller Cycle Count for Major Routines 5-80
Figure 6-1. Configuration File Format for the IC Utility (3 sheets) 5-83
Figure 6-2. Interconnect Utility Screens (2 sheets) 5-85

1.0 INTERCONNECT ADDRESS OF SPACE ON MULTIBUS II

1.1 Who Should Read This Document?

The primary audience of this document consists of companies and individuals who are in the process of designing their own Multibus II boards for use with other compatible products. It is assumed that the reader has already studies the Interconnect Interface Specification and has a good working knowledge in the operation of interconnect space on existing Intel products. In addition, portions of the IEEE 1296 specification and the MPC User's Guide are referenced in some detail. A complete bibliography of recommended reading material is contained in Appendix A.

This design package consists of two related items. The first item is the document you are now reading which is a user's guide to the overall design process. The second item, is a diskette containing copyrighted software to be used in generating new firmware for your interconnect subsystem. This package is not intended for users to change the content of microcontrollers already installed on Intel boards.

1.2 Configuration Issues in Microcomputer Based Systems

Over the past few years, microcomputer designs have progressed dramatically in capability and performance. In contrast, little progress has been made in enhancing ease-of-use. Until recently, board users have had to deal with the added complexity of modern single-board computers by wading through lengthy reference manuals and innumerable jumper options—often arriving at the final solution only by trial and error. System integrators often found that the firmware revision number penciled in on the EPROMs they installed did not match the device driver revision and consequently, nothing works! Worse yet, things might work for a little while and then fail; resulting in wasted time debugging the problem to determine what went wrong. Memory mapping options, arbitration priorities, interrupt levels, and scores of other "tunable" parameters contribute to the fray, leaving the system designer befuddled and confused. Often, the only way out of this mess was to locate a board that was already properly configured and copy off the jumper list.

Board designers build in numerous options so their products can be used in the broadest possible spectrum of applications. The number of options offered is not the core of the problem, but managing them is. Multibus II addresses this problem with a special address space known as "interconnect". Now for the first time, system-wide configuration information has been made accessible to software; thereby opening opportunities for centralized control and coordination. In most cases, the end user of these products will be completely unaware of the configuration process. They simply remove the board from its shipping container, install the proper firmware, plug it into a free cardslot in the backplane, and apply power. Things work the first time around with no mess, no fuss, and no configuration errors.

Interconnect is great for end users; it eliminates most of the common configuration errors, speeds up the installation process, and facilitates diagnostics and repair. When considered in the context of an overall system architecture that includes message passing, interconnect is one of the foundation building blocks distinguishing Multibus II as an environment capable of satisfying the most demanding of applications.

1.3 Overview of Interconnect Address Space

Interconnect address space is a fundamental part of the IEEE 1296 specification, which defines Multibus II. Interconnect address space was included in the IEEE 1296 specification to solve three major problems: board identification, configuration, and diagnostics. The board identification registers are read-only locations containing board information such as type, manufacturer, components installed, and other board specific functions. The configuration registers are read/write registers which allow the system software to set and change the configuration of many on-board hardware options. In most cases, hard-wired jumper options can now be eliminated in favor of software control. The diagnostic registers are used for the starting, stopping, and status reporting of self-contained diagnostic routines supplied with each board. These diagnostics are commonly known as Built-In Self Tests (BISTs).

1.3.1 GEOGRAPHICAL ADDRESSING

Interconnect is based on the fundamental principle that you can locate boards within a backplane using a system of cardslot numbering. This concept, known as geographical addressing, is a very useful tool during system-wide initialization. Each board in the system contains firmware which conforms to a standardized header format (Figure 1-1).

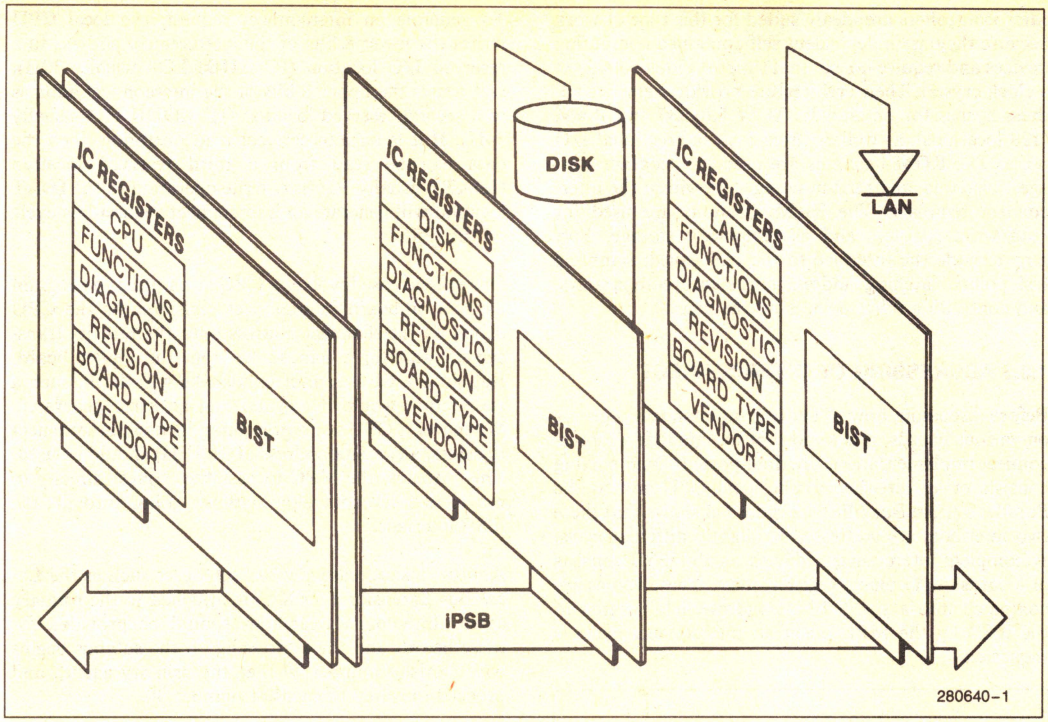


Figure 1-1. Interconnect Overview

5

At boot time, the system software will scan the backplane to locate its resources before loading in the device drivers. This approach eliminates the need for reconfiguring the software every time a new board is introduced into the backplane. It also solves the problem of how to configure multiple instances of controller and processor boards in large multiprocessing systems. Cardslot independence is achieved by having all boards in the system carry their own initialization and diagnostic functions on-board in firmware. Operating systems can generate a map of where resources are located during initialization and then use this map as a base address list for message passing.

1.3.2 MICROCONTROLLERS IN THE BUS INTERFACE

Most Multibus II designs are based on a highly integrated bus interface controller known as the Message Passing Coprocessor (MPC). Special provisions have been made in the bus interface silicon to enable board designers to implement interconnect in a cost effective manner. A typical Multibus II interface consists of the MPC, a small number of bus transceivers, and a microcontroller such as the Intel 8751 or equivalent (Figure 1-2). It is the microcontroller (in association with the MPC) that has the responsibility for all interconnect functions.

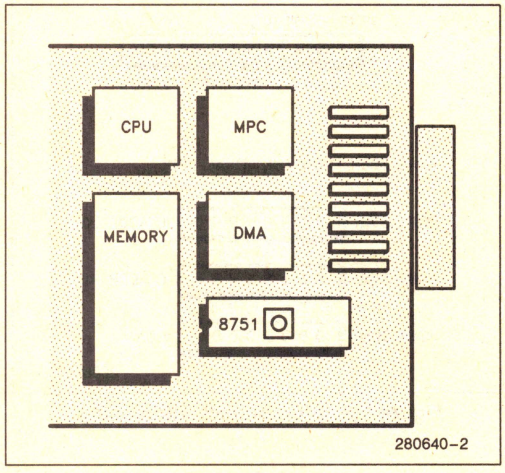


Figure 1-2. Hardware Support for Message Passing

Microcontrollers are idealy suited for this type of work because they are independent self-contained computing devices and require no external support chips outside of a clock crystal. Their architecture provides separate address spaces for on-chip ROM (4 Kbytes) and RAM (128 locations), as well as three 8-bit bidirectional I/O ports. The ROM locations are used for program storage, constants, and read-only registers within the interconnect template. The RAM locations are used for read/write registers and as temporary storage. Port pins provide the interface to the real world; sampling test points, latching address terms into comparators, and controlling other devices on the board.

1.3.3 ADDRESSING OF INTERCONNECT

Before discussing how to address interconnect registers on various boards, it is important to note that all interconnect implementations are dual-ported. Dual porting consists of an interface to the local CPU and to the Parallel System Bus (iPSB). Figure 1-3 shows that these two interfaces are addressed in slightly different ways. A complete interconnect address on the iPSB consists of a cardslot ID plus a register offset. These values are combined into a single 16-bit address field written to the iPSB by the MPC when an interconnect cycle is requested.

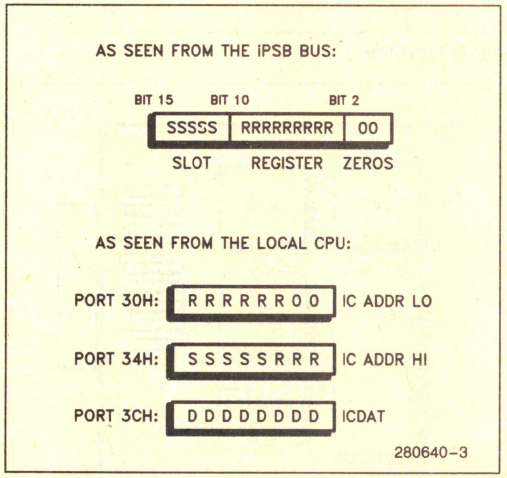


Figure 1-3. Interconnect Addressing

To generate an interconnect request, the local CPU writes the lower 8 bits of the interconnect address to a reserved I/O location (IC ADDRLO—usually 30H), and writes the upper 8 bits of the interconnect address to a second reserved location (IC ADDR HI—usually 34H). If it is an interconnect read operation, then the data can be read from a third reserved location (IDAT—usually 3CH). A write operation to the IDAT location will generate an interconnect request bus cycle on the iPSB.

One special case involves a CPU attempting to program its own on-board interconnect registers. As the CPU drives an interconnect address onto the bus, its transceivers wait for a handshake from the replier board. But since a CPU cannot handshake with itself, such a transaction would be invalid and an error generated. Whenever a CPU is programming its own interconnect registers, a cardslot address of 31(1FH) should be used. This instructs the MPC to pass the request directly to the local microcontroller without going through the iPSB interface.

Another special case is when sub-buses such as the Local Bus Extension (iLBX), are attached to the primary agent. Interconnect facilities should be provided for these boards and the addressing on the sub-bus begins with cardslot number 24 (i.e., the primary agent), and proceed upwards to cardslot number 30.

1.3.4 DATA STRUCTURES IN INTERCONNECT

The objective of interconnect address space is to allow higher level software to gain information about the environment in which they operate independent of who manufactured the board, what functions it contains, and what cardslot it resides in. To accomplish this goal, an Interconnect Interface Specification has been published and forms the basis for much of the information in this guide. If you have not yet read this document, you should do so before beginning your design effort.

Interconnect functions implemented on Intel's single board computers go beyond the requirements of the IEEE 1296 specification. This specification mandates that all conforming products include an Interconnect Header Record. The header record consists of information regarding board type, its manufacturer, what firmware is installed, and other relevant information. An example header record is shown in Figure 1-4. In addition to the header record, the manufacturer may also supply additional function records which make other features of the board accessible through interconnect.

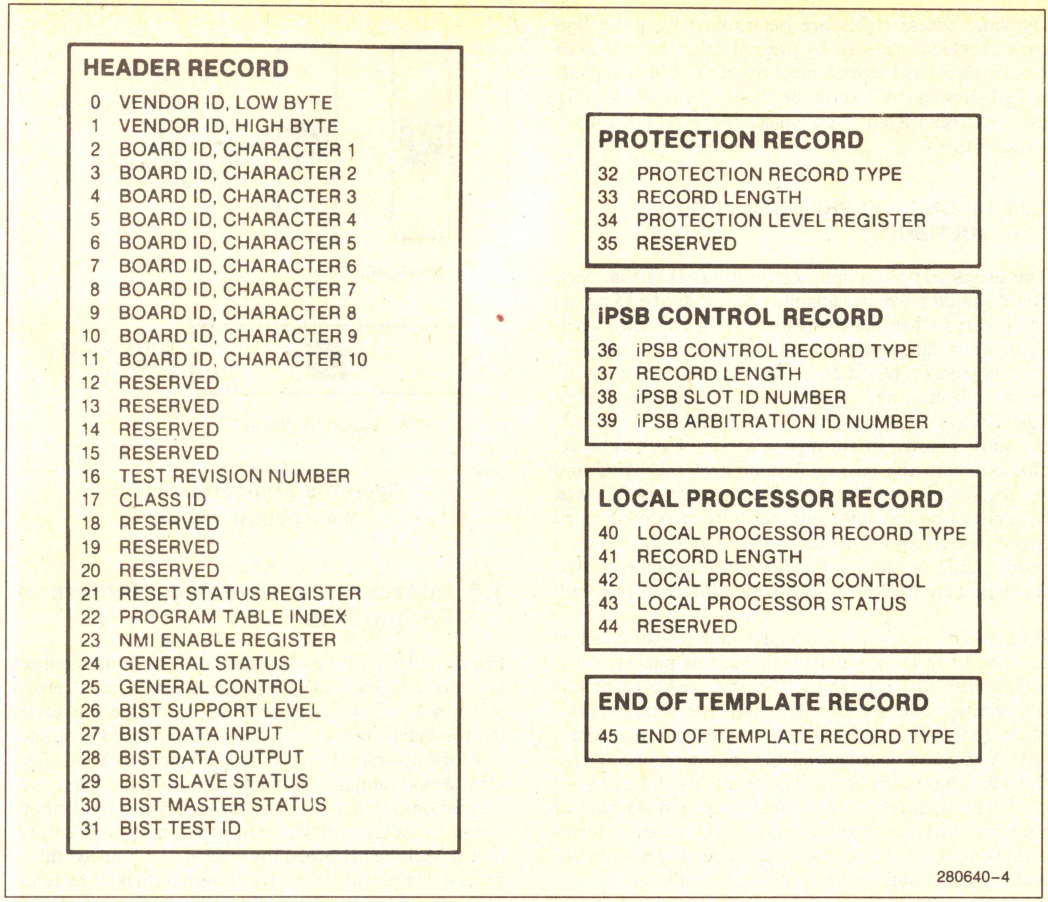


Figure 1-4. Function Records in the Interconnect Core Firmware

Function records begin with a byte specifying the record type followed by the number of bytes that make up the record. The manufacturer must publish a description of these function records in their reference documentation. Many types of function records have already been defined. Some examples include memory configuration, parity control, serial I/O, and other commonly used functions. If the existing record types do not adequately describe a function, a new record type can be defined—up to a maximum of 1020 different record types are allowed.

The system software initiates the search for a specific function record at interconnect register 32, which is the first function record following the header record. The program first checks the record type field and then counts bytes to the next record type field until either the correct function record is found or the End of Template (EOT) record (hex value 0FF) is encountered.

1.3.5 ACCESS RIGHTS AND PROTECTION RECORDS

Because the interconnect registers are dual-ported, each has a set of static and dynamic access rights that detemine which operations will be allowed on either the local or iPSB bus interface. All interconnect registers can be read from either bus, however, static access rights may place restrictions on whether a register can be written to from either interface. The term "static" is used because these access privileges are predetermined by the designer of the interconnect firmware and will not change during system operation. In other situations it may be desirable to allow a register to be modified during system initialization; then locked against further changes during normal system operation. This capability is essential since many of the functions contained in interconnect are so vital to correct system operation that some means of protecting them from malicious or inexperienced users is required.

Dynamic access rights are determined by protection records which are used to prevent other boards from modifying a local interconnect resource. When activated, all subsequent records become read only so other users can read from interconnect registers, but cannot write to them.

1.3.6 DIAGNOSTIC PHILOSOPHY OF MULTIBUS II

The diagnostic philosophy of Multibus II is that each board should have the capability to test itself and report error status when problems exist. There are two occasions when diagnostic testing is invoked. A subset of the complete on-board diagnostics is run during power-on initialization and more extensive testing can be invoked from the operator's console. Following power-on, most boards go through a series of initialization checks where the basic functioning of the MPC and microcontroller are verified. Initialization is followed by a power-on test suite automatically invoked by each board. If a hardware failure is detected at this point, a yellow LED on the front panel will illuminate so that the failing module can be easily identified and replaced.

If further testing is desired, extended diagnostics can be invoked by placing a diagnostic request packet in the interconnect BIST registers. Usually one board acts as the Master Test Handler and requests services from other system boards functioning as Slaves when under test. A menu of tests is available via interconnect. This test philosophy can be applied on-site by the end-user, service representative, or remotely executed via modem from the regional repair center. In most cases, downtime is minimized by sending out a replacement board and thus avoiding an expensive repair call.

The firmware content of Multibus II boards is much greater than that found on previous industry standard buses. In addition to the 8751 Microcontroller, Multibus II boards normally host EPROMs that contain extended diagnostics (BISTs), test handlers, reset initialization sequencing, debug monitors, and many other functions. The location of diagnostic firmware on a board (Figure 1-5) is dependent on code complexity and execution speed. For simple replier agents, the microcontroller's on-board EPROM may have enough program storage space for diagnostic functions as well as the interconnect firmware. In contrast, the majority of the requester/replier boards (i.e., capable of becoming bus masters), are more complex and most diagnostic code is run on the CPU from on-board EPROM. In this case, the microcontroller primarily serves as the communication interface for the diagnostics.

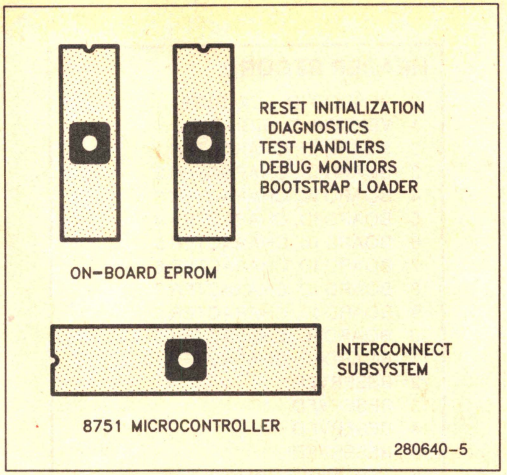


Figure 1-5. Firmware Content of Multibus II Boards

1.4 Interconnect—The Manufacturer's Perspective

From the perspective of a board designer, interconnect is a mixed blessing. The board manufacturer is certain to enjoy the benefits of reduced support costs, easier fault isolation in field repairs, and enhanced customer satisfaction—but these advantages do not come free. One would anticipate longer development time, increased on-board part count, and the firmware configuration to increase the amount of effort it takes to prepare a Multibus II board for market. And indeed this is so. If a competent design team were to tackle the interconnect subsystem (including diagnostics, hardware, and firmware design), it would not be unreasonable to allow six man-months for the job.

Given that this represents an extraordinary investment for the manufacturer, the primary goal of this design package is to reduce the amount of time required to include interconnect in your design from six months to only six days! Another goal is to guarantee compatibility and interoperability of your products by placing common core functions in user extensible firmware in such a way that it is easy to customize the design to fit your own particular needs.

1.5 Assumptions Regarding Your Operating Environment

Although many implementations of interconnect are possible, it was necessary to restrict the scope of this

guide to satisfying the broadest and more typical range of users—designs based on the MPC component in association with an 8751 Microcontroller (or equivalent). This design guide is intended as a generic solution that meets the needs of most of these users.

For software development, it will be necessary for you to obtain an IBM PC (model XT, AT, or compatible) plus the appropriate Intel programming languages and a PROM programmer to transfer your code into the microcontroller. An in-circuit emulator is not required for this project; however, some users will find it expedient to make use of such a tool since it simplifies debugging and eliminates the need for PROM programming while the code is being developed. A complete list of hardware and software requirements may be found in Section 2. Any departure from the recommended development tools or practices is outside the scope of this document and may lead to unpredictable results.

1.6 Design Methodology—An Overview

The process of designing an interconnect subsystem invariably begins with a high level discussion of what function you intend to support. While the header record is quite easily defined, decisions as to what function records to include should be carefully considered in terms of how much flexibility to give your users, what functions they might be interested in, and how much external hardware will be required. Some of the function records listed in the Interconnect Interface Specification are already implemented in the core firmware and require minimal effort to support. Other functions may be quite complex and could potentially require extensive TTL circuitry external to the microcontroller. As with all engineering designs, you should spend a significant portion of your time making sure that you have a clean workable specification before proceeding into the implementation phase.

The second step in interconnect design is to determine what circuitry is required to gain access to the information in interconnect that you intend to present to the user. This consists of locating all test points, control circuitry, latches, and transceivers external to the microcontroller. Most likely this determination will be made at a point where the overall schematic for the board is near completion and before you begin the layout and develop the prototype. At this stage, all dedicated I/O addresses will be defined and rudimentary PAL equations for the control points will be writen.

The third step of interconnect design consists of evaluating your on-chip resource requirements based on the function records being implemented. At this stage, you will write the functional routines and identify the RAM, ROM, and port requirements for the microcontroller. If your original interconnect specification was overambitious or in appropriately defined, you will discover at this point that you may be forced into external PROM, static RAM, or port expansion logic; and may wish to scale back your design or change over to the 8752 Microcontroller (having 8 Kbytes of ROM and 256 bytes of on-chip RAM). Once you know your resource requirements and have written the functional routines, you are now ready to integrate your custom code with the core interconnect firmware.

The fourth step in the design process consists of loading the tables with data based on the interconnect template you specificed in step one, plus the external declarations for the routines you wrote in step three. The object code supplied on the diskette with this guide contains a table driven collection of routines that provide the core interconnect functions. These give the user some commonly used function records (Figure 1-4) and provide the opportunity for users to add their own routines to this core. Generating the firmware consists of assembling your code and then allowing the table generators to integrate this code into the core module through an ASM-51 macro expansion process. Once complete, the entire package is integrated using RL51 (a relocation linker) to resolve any external references and produce a unified object module for loading into the microcontroller EPROM.

The final step in the process is to program the microcontroller and test every imaginable function and event sequence within interconnect. It is at this stage that the use of an in-circuit emulator, logic analyzer, or oscilloscope may be desirable to help in localizing logic faults or timing related problems. In most cases, debug time is fairly short since the core routines are supplied already and are known to be good.

Once the interconnect subsystem is totally tested, the board can be forwarded to the device driver development team and/or system integrator for initialization software development and further functional testing.

2.0 PREPARATION FOR USE

2.1 Hardware and Software Requirements

Before proceeding with your interconnect design using this guide, the hardware, software, and optional equipment listed below are required (at a minimum).

Hardware Requirements:

- IBM PC (model XT, AT, or compatible) configured with at least 640 Kbytes of internal memory and a 10-Mbyte (or larger) hard disk.

- Intel PROM Programmer, model iUP201, plus the 8751 Microcontroller Personality Module and a serial cable. The IBM PC must have a spare serial port to interface with the PROM Programmer. (Note that other brands of PROM Programmers can be used, but the batch files and object module produced by Intel's development tools are not guaranteed to be compatible.)

Software Requirements:

- DOS Operating System, version 3.0 or greater
- iPPS PROM Programming Software, version 2.2 or greater
- ASM-51 Macro Assembler version 2.2 or greater
- RL51 Relocation/Linkage package version 3.0 or greater

Optional Equipment:

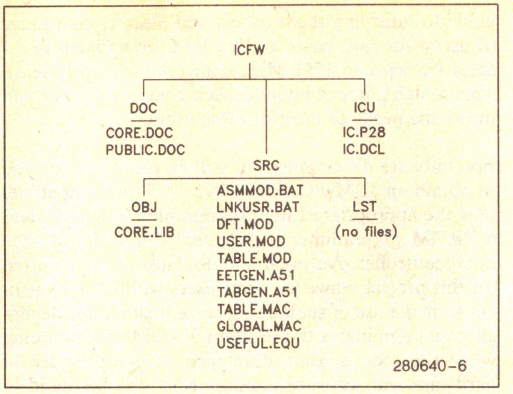
- ICE™ 51 (or ICE 5100) In-Circuit Emulator with IBM PC Interface Card
- Oscilloscope

2.2 Loading Software onto Your System

Before starting, it is assumed that you will have already loaded the ASM-51 and RL51 program files in a common subdirectory of the hard disk and have indicated the route to that subdirectory in a PATH command entered into the AUTOEXEC.BAT file (in the root directory). This will allow you to call these program files from any point in the file structure without having to identify the directory search path to these files.

Install the software supplied with this design guide onto your hard disk by inserting the interconnect firmware diskette into drive A or B. After the C> prompt, type A (or B): to change the default drive. Next type INSTALL and then press **Enter.** The computer will read from the drive specified and immediately start executing the install.bat batch program. When this batch program completes processing, you will find the following added to the subdirectory structure of your hard disk:

Batch files are supplied to automate the firmware generation process. These files make some assumptions about your hard disk directory structure. It is important to note that all batch files must be called from the \ICFW\SRC directory path, otherwise the DOS command processor will not look beyond your current directory in its search for a file name. Edit the path command in your AUTOEXEC.BAT file to include the \ICFW directory.



2.3 Interconnect the Easy Way

While interconnect is a complex topic, there is an easy way to get a functional interconnect subsystem operational without detailed knowledge of the internal design. To do this, view the file \ICFW\SRC\TABGEN.A51 using the TYPE command or a text editor and notice that the data fields for vendor ID, board ID, hardware test revision, and class ID have been left blank (looking ahead, this is Figure 5-1). Consult the Interconnect Architectural Specification to determine what information to place in these fields. Once you have obtained this information, perform these steps at your computer console:

a. Type CD and then press **Enter** to display the current directory. If \ICFW\SRC is not being displayed, then type CD\ICFW\SRC and press **Enter** to change to the correct directory path.

b. Run the following batch programs in the order listed:

```
ASMMOD  dft
ASMMOD  user
ASMMOD  table
LNKUSR  test
```

The result is a PROMmable object code file (TEST.LNK) that is placed in \ICFW\SRC\OBJ subdirectory. The TEST.LNK object code is ready to burn into the microcontroller EPROM and provides a complete interconnect header record as well as protection, iPSB control, and local processor records.

The core hardware design consists of the minimum interconnect implementation as shown in Figure 4-1. The basic combination of hardware and firmware can be used during prototyping as the starting point for most interconnect designs.

3.0 THEORY OF OPERATION

3.1 MPC to Microcontroller Hardware Interface

Most Multibus II designs use the Message Passing Coprocessor (MPC) component with an 8751 Microcontroller to implement the Parallel System Bus (iPSB) interface. This combination minimizes the number of devices required to implement a full-featured bus interface and provides flexibility in adapting the design to the broadest possible range of functional specifications. The hardware interface between the MPC and the microcontroller is shown in Figure 3-1.

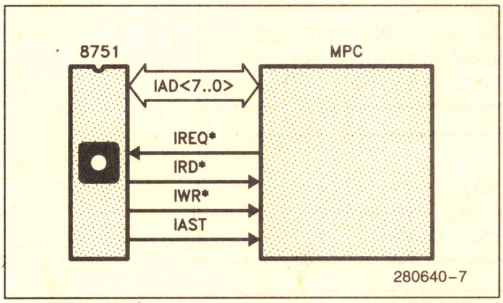


Figure 3-1. MPC to Interconnect Pathway

The MPC is designed to sit directly on the microcontroller's multiplexed Interconnect Address/Data bus (IAD<7..0>). When an interconnect cycle is initiated, the **IREQ*** signal from the MPC interrupts the microcontroller with a request for services. The microcontroller then performs a series of read and write operations to a group of MPC interconnect bus registers to satisfy the interconnect request and complete the operation. In terms of hardware control, the microcontroller acts as the bus master on this interface; generating the read and write signals, and supplying an Interconnect Address Strobe (**IAST**) based on its own Address Latch Enable (**ALE**) signal. The reader is requested to review Sections 4 and 5 of the MPC User's Manaual before continuing further.

3.2 MPC Interconnect Bus Registers

The MPC component contains a set of special function registers that are only accessible via the IAD bus. These registers (Figure 3-2) can be categorized into five functional groups: interconnect reference registers, slot and arbitration ID registers, configuration registers, diagnostic registers, and the no access registers.

The MPC interconnect reference registers serve as the basic communications interface between the microcontroller and the MPC. Whenever the local CPU or iPSB agent generates an interconnect request cycle, the registers actually being accessed (IC ADDR HI, IC ADDR LO, IDAT) physically reside in the MPC rather than in the microcontroller. The MPC asserts the **IREQ*** signal to interrupt the microcontroller which responds by initiating a dialogue of read/write commands to the MPC interconnect reference registers.

The second functional group of MPC interconnect registers are concerned with the cardslot and arbitration ID assignments made by the Central Services Module (CSM) during reset initialization. Note that the Interconnect Interface Specification describes an iPSB Control Record which includes registers for both arbitration and cardslot ID. This allows a CPU to determine in what cardslot it is residing.

The third functional group of MPC interconnect registers control configurable features on the MPC such as dual-port address boundaries, arbitration priority, Reset-Not-Complete (RSTNC) control, error reporting, and fail-safe counter functions. In most implementations, these registers are passed through an interconnect function record to make them user accessible and configurable.

The fourth functional group of MPC interconnect registers control retry and diagnostic functions. These parameters can be used for performance tuning and confidence testing, but are not frequently accessed by users. In general, they can be programmed during initialization with default values and then ignored.

Finally, there is a group of addresses between 0 and 7FH for which the MPC guarantees a tristate condition (no access). Interconnect designers can take advantage of this feature by using these addresses to decode registers and latches residing directly on the IAD bus. This technique provides a convenient way to generate an 8-bit bidirectional bus using user defined addresses as chip enable terms. The reader is encouraged to read all of Section 7 in the MPC User's Guide before proceeding.

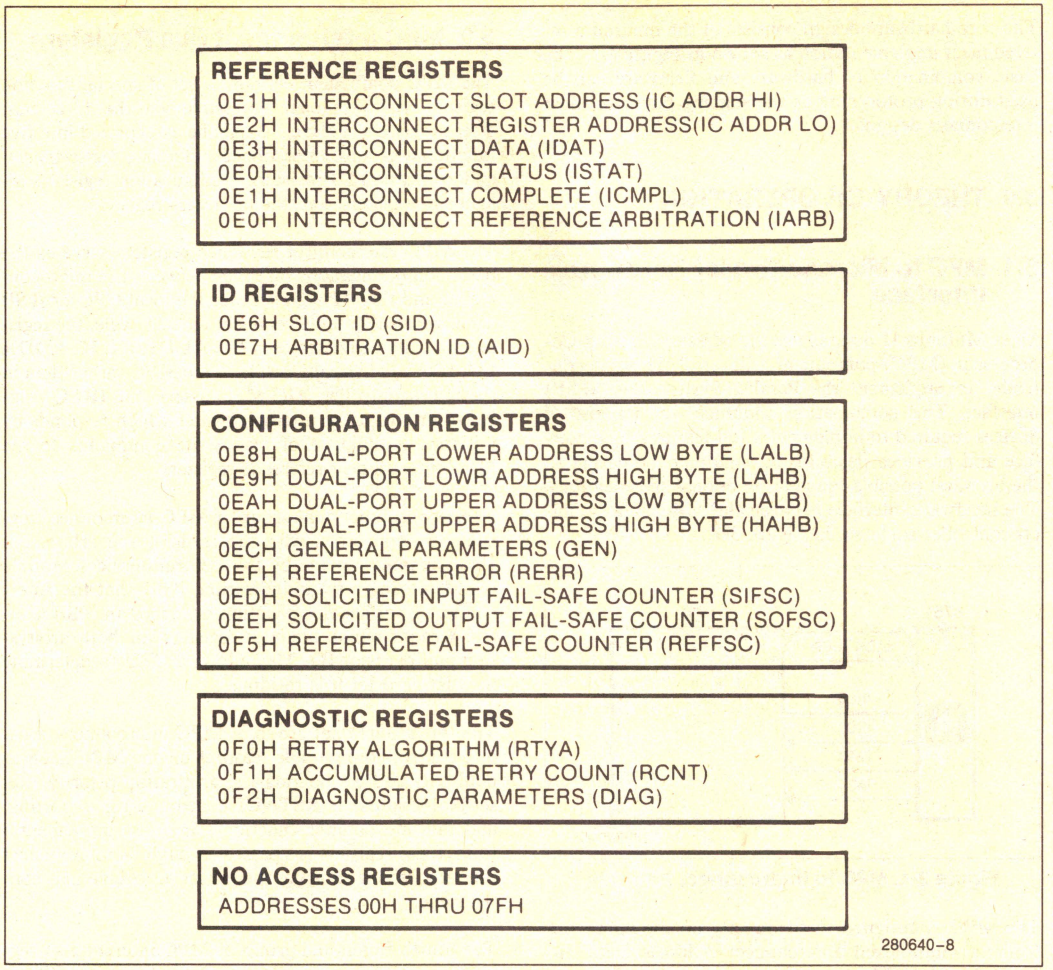


Figure 3-2. MPC Interconnect Registers

3.3 Partioning of Functions between the Microcontroller, CPU, and MPC

Because of the close association of the CPU and the microcontroller to the MPC, a number of interdependencies arise—especially during initialization and diagnostic testing. Some of the more complex functions such as dual-port memory control and fail-safe counters actually cross component boundaries. Thus, one must view the complete interface as a functional subsystem. The following discussion highlights the more important interdependencies regarding the interconnect subsystem.

3.3.1 MPC DIAGNOSTIC TESTING

A special feature of the MPC is its ability to simulate message passing operations to itself without presenting data to the iPSB bus interface. This mode is termed "MPC Loopback Testing" and is enabled by the iPSB Diagnostic Register in the iPSB Control Record. This causes the microcontroller to set the Reset-Not-Complete Out (RSTNCOUT) bit in the MPC General Parameter Register. The result is that any messages the local CPU loads into the MPC transmit buffers are routed directly to the MPC receive buffers. While this is happening, the MPC Buffered Address/Data bus

(BAD<31..0>*) is active and it is necessary for the microcontroller to tristate the iPSB buffer logic by preventing the iPSB Transceiver Output Enable (**BTROE***) signal from going active low. Note that MPC loopback testing is only allowed while **RSTNC*** is being asserted by the host agent.

3.3.2 DUAL-PORT MEMORY CONTROL

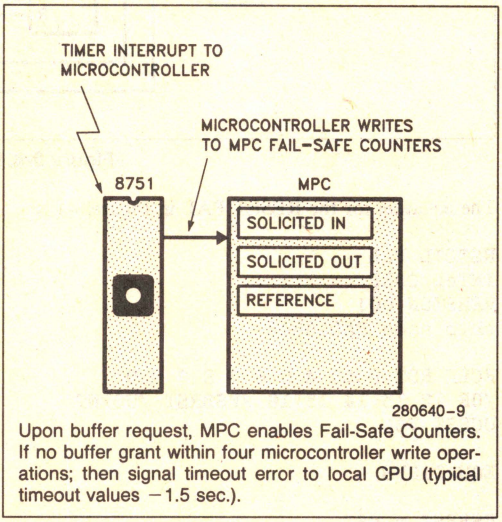
When another iPSB agent selects your board to participate as a replier in a memory reference, the address recognition function for dual-port memory is performed by the MPC based on the starting and ending addresses programmed into the MPC interconnect registers during initialization. When an address match is found, the MPC will drive the **SEL*** signal to your dual-port memory controller and wait for a **COM*** or **ERR*** signal to be returned before completing the cycle. During this transaction, the MPC provides all parity generation and checking, system control, and wait-state signal generation services to the iPSB bus interface. Valid address selection may occur on any 64-kByte boundary within the 4-Gbyte memory space; however, it is advisable to include value checking in your microcontroller firmware to ensure that the user doesn't enable more memory than is physically present on the board. Only one bank of contiguous memory is supported when using the MPC dual-port functions.

3.3.3 MESSAGE RETRY OPERATIONS

When message traffic is arriving faster than the local CPU can receive it, the MPC FIFO buffers will overflow and some form of flow control must be initiated. The MPC has a special retry mechanism for this condition that is controlled through registers accessible to the microcontroller. Retry is enabled by setting a bit in the MPC Diagnostic Parameters Register and the delay between retries is selected in the Retry Algorithm Register. The Accumulated Retry Count Register indicates how many Negative Acknowledge (NACK) errors have occurred for a given message attempt. The Accumulated Retry Count Register is used with the Retry Algorithm Register to tune system performance by selecting the most effective retry interval.

3.3.4 FAIL-SAFE COUNTER FUNCTIONS

When the MPC issues an iPSB buffer request, there is no guarantee that a buffer grant will be returned in a reasonable amount of time. If enabled, the MPC Reference Fail-Safe Counter will cause an error interrupt if no buffer grant is received by the end of a timeout period (typically 1.5 seconds). Likewise, a similar fail-safe timeout exists for reference operations in the unlikely event that they are unable to acquire the iPSB bus due to arbitration or Bus Clock (BLCK) problems. In either case, the microcontroller acts as a programmable timebase by writing to the MPC fail-safe counter addresses on a periodic basis in response to an internal timer interrupt (Figure 3-3). When the MPC starts the buffer request or reference operation, it enables the fail-safe timeout and waits for the operation to complete. If the microcontroller is able to write to a MPC fail-safe counter address four times before the bus cycle completes; then a timeout interrupt is asserted to alert the CPU to the problem. Note that fail-safe counter functions should be disabled during debugging since breakpoints set by human intervention may prevent the MPC from completing an operation before a timeout occurs.



Upon buffer request, MPC enables Fail-Safe Counters. If no buffer grant within four microcontroller write operations; then signal timeout error to local CPU (typical timeout values −1.5 sec.).

Figure 3-3. MPC Fail-Safe Counters

3.4 Multibus II Reset Conditions

In a Multibus II backplane, the **RST*** signal is used for the system-wide reset. Additionally, the **DCLOW*** signal designates power-fail indication and the **PROT*** signal designates an early warning battery back-up control. One of the options available to the board designer is to use combination logic in association with the microcontroller to further define three categories of reset conditions: cold-start, warm-start, and local reset. In Figure 3-4, the reset circuitry that gives a board the capability to distinguish between these events is shown. The reset control logic (in the PAL16R4B) signals a cold-start whenever reset is accompanied by a low power condition and signals a warm-start in all other cases. Local resets are generated by an interconnect operation to the microcontroller, which then pulses the CPU reset line.

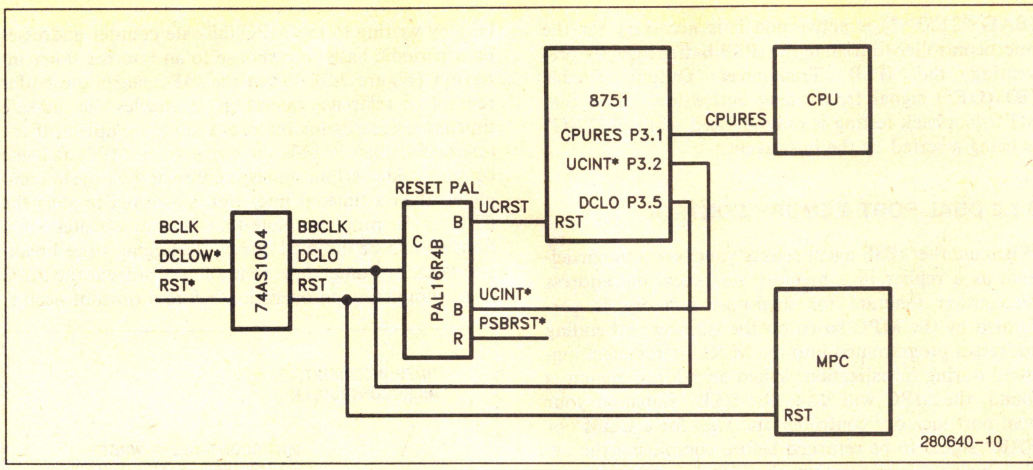


Figure 3-4. Reset Hardware

The equation for the RESET PAL in Figure 3-4 is:

```
RSTCTL PAL
INTeL CORP
VERSION 001
chip name PAL16R4B

BCLK RST DCLO 4 5 6 7 8 9 GND
/OE 12 13 14 15 16 /PSBRST /UCINT
UCRST VCC

EQUATIONS

PSBRST = RST
UCINT = PSBRST * /DCLO * /UCRST
/UCRST = /PSBRST + (/DCLO * /UCRST)
```

3.4.1 COLD-START AND RECOVERY RESET

A cold-start condition represents the state of the backplane following a power-on sequence, while the recovery reset represents the state of the backplane following power failure situations. The CSM will cycle through its normal sequence of assigning cardslot and arbitration IDs during a 50 ms period while the **RST*** signal is active low. The iPSB reset signal is buffered in on each agent and then routed directly to the MPC and the microcontroller. When reset is released from the iPSB, the microcontroller will force a reset to the local CPU and then execute some rudimentary BISTs (initialization checks). The interconnect template will be loaded with default values and the MPC gets initialized to its default state. The CPU is then released from reset to execute its power-on BISTs (some of which are likely to alter the contents of memory by overwriting data previously present).

In the firmware supplied with this guide, the CPU must clear the RSTNC* bit in interconnect within 30 seconds after power-on or else the microcontroller will again assert reset to the CPU on the assumption that something is seriously wrong with the board. (This is done to ensure that the failure of a single system board will not prevent the remainder of the system from coming alive due to **RSTNC*** being held active low.)

3.4.2 WARM-START RESET

A warm-start condition is normally the result of a human operator pressing a front panel reset button or programming the CSM for a system-wide reset via interconnect. The sequence of events is similar to the cold-start except that memory will remain unchanged, BISTs are not run, and all interconnect configuration registers remain unchanged. Warm resets are very useful for recovering from a hung system condition during debugging because the system cycles through reset much more quickly and retains all configuration and error status information.

3.4.3 LOCAL RESET

Local resets selectively return all circuitry behind the bus interface on a single agent to a known condition without effecting any other system boards. Local reset is generated through interconnect programming and does not disturb the contents of local memory or the interconnect configuration registers. The microcontroller BIST diagnostics are not run and the agent returns quickly to a known condition. Local resets are often used to recover from a partial system failure condition caused by improper programming or single agent failures.

There are two types of local reset supported: self-toggling and non-toggling. The Interconnect Interface Specification describes a local reset control (bit 7) of the General Control Register in the Interconnect Header Record of each board. This is a non-toggling reset that holds the agent inactive until specifically cleared by further interconnect programming. This feature allows an external agent to clamp reset to that board, thereby taking it off-line until further notice. Note that the board will not respond to other resets (warm or local) until this bit is cleared. This type of local reset is not suitable if the intent is for the agent to reset itself since the board is incapable of returning from the reset condition to clear the bit.

For this reason, many Multibus II boards offer a self-toggling reset control in the Board Specific Record that will clear itself soon after being set (typically 20 ms). One might be tempted to use this register for all local resets in preference to the General Control Register, however, it is not supported on all Multibus II boards and its absolute location in the interconnect template will depend on how many other function records precede it. In common practice, the local reset bit in the General Control Register is a better solution in all cases except self-toggling resets.

3.5 Initialization of the Microcontroller following Reset

Immediately following a cold or recovery reset, all bus interface components must be initialized to a known state. The microcontroller is normally the first to undergo initialization since it will be released from reset before the CPU and has the added responsibility for the MPC interconnect registers. Flowcharts of the program flow in the microcontroller are provided in Appendix B.

The first priority of the microcontroller is to establish its own environment by creating a stack and loading the Special Function Registers (SFR's), which control the on-chip resources such as the ports, timers, and interrupt priorities. Next, the microcontroller initialization checks are run as a basic "sanity check" of itself. There are three BISTs in the initialization sequence:

a. Microcontroller RAM—A simple read/modify/write test to on-chip RAM locations.

b. Microcontroller ROM—A check sum test on the on-chip ROM.

c. IAD Bus and MPC Accessibility—A "walking ones" pattern is written to several MPC registers and read back to verify functionality of the MPC interconnect registers and the IAD bus.

The interconnect registers that reside in RAM (i.e., those that are R/W), are now loaded with their default values. These default values are copied from a ROM-based default table (the DFTABLE), specified at configuration time. Next, the microcontroller on-chip timer/counters and the MPC interconnect registers are initialized. If the user has any unique initialization requirements, a call to the INIT_USER procedure is made at this time. Up to this point, the local CPU has been held in reset and the iPSB transceivers are being held tristate. Now all of these resources are released and timer/counters startup and the CPU has 30 seconds to clear the RSTNC* bit in interconnect. The microcontroller then enters the mainline routine of the firmware.

3.6 Interrupts Driven Firmware Operations

Figure 3-5 illustrates the various interrupt sources which drive the interconnect functions on the microcontroller. The mainline code consists of very simple polling operations that execute with interrupts disabled. These operations consist of polling for the Non-Maskable Interrupt (NMI), BIST, and reset conditions, plus updating the front panel LEDs. At the end of each pass through the mainline code, interrupts are enabled prior to executing the jump instruction that starts the next pass through the mainline code.

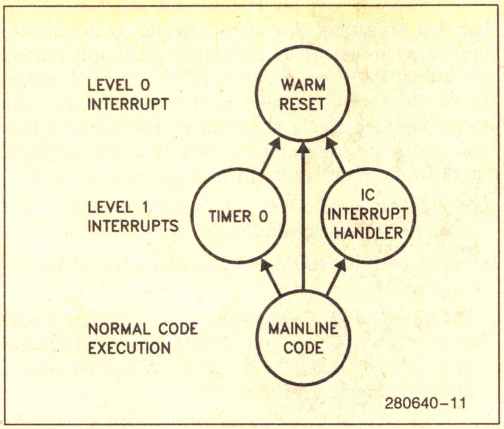


Figure 3-5. Microcontroller Interrupt Sources

Nearly all of the important operations within interconnect are interrupt driven. These include the IC interrupt handler (which services the MPC **IREQ*** signal), a reset interrupt handler, and a group of timer based functions. A general description of these operations follows.

3.6.1 INTERCONNECT INTERRUPT HANDLER

The IC interrupt handler is called whenever the MPC signals to the microcontroller that a request for interconnect service is pending (**IREQ*** has been asserted). The microcontroller responds by reading the MPC's interconnect reference registers to determine which register in interconnect has been requested; whether this is a read or write operation; and whether this is the local CPU or an iPSB agent generating the request. If the register number is beyond the End of Template (EOT) record; then no error status is posted and the interconnect operation is terminated. Otherwise the microcontroller begins a table driven look up of that register and completes the interconnect cycle.

3.6.2 INTERCONNECT READ OPERATIONS

Interconnect read operations are relatively simple because no need exists for checking the access rights or data values (see Figure 3-6). The register number (REGNUM) is used as an 8-bit offset into the read vector table (RVTABLE), which then supplies an offset into the read jump table (RJTABLE) containing a long jump instruction to the address of the read routine. This indirect look up technique is used to allow several different registers to share the same read routine and to improve code compaction. Indirect jump tables also allow the user to add their own read routines to the template following the interconnect core functions.

Once the read operation is complete, the results are available to the IC interrupt handler in the ICDATA register. The data is copied to the MPC IDAT register and completion status is posted. From the IC interrupt handler, control passes back into the mainline code. Note that interrupts were enabled within the IC interrupt handler (just prior to executing the long jump instruction), and that the reset interrupt has a higher priority.

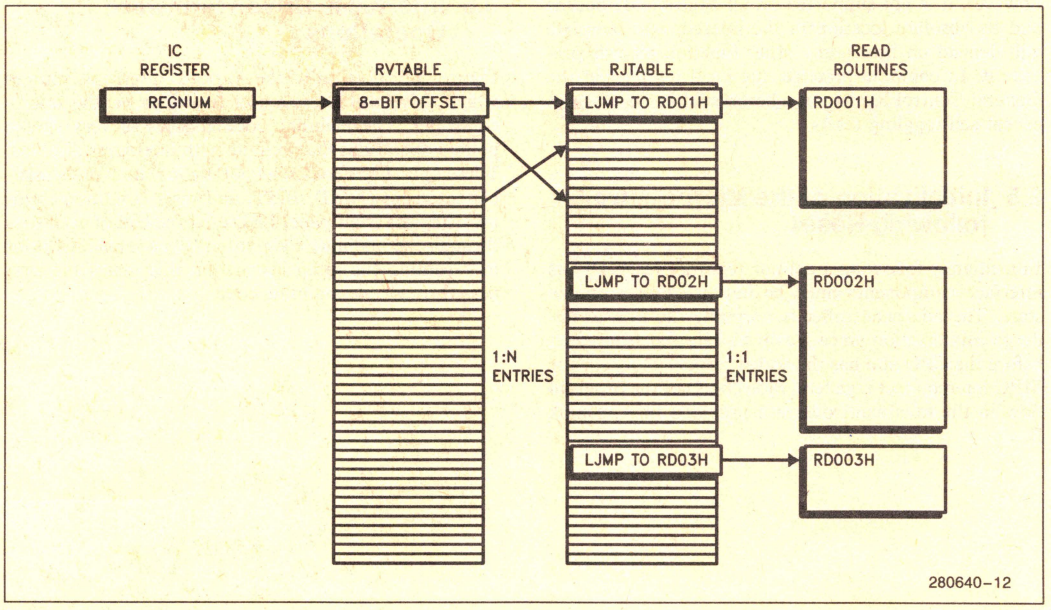


Figure 3-6. Indirect Jump Tables for Read Routines

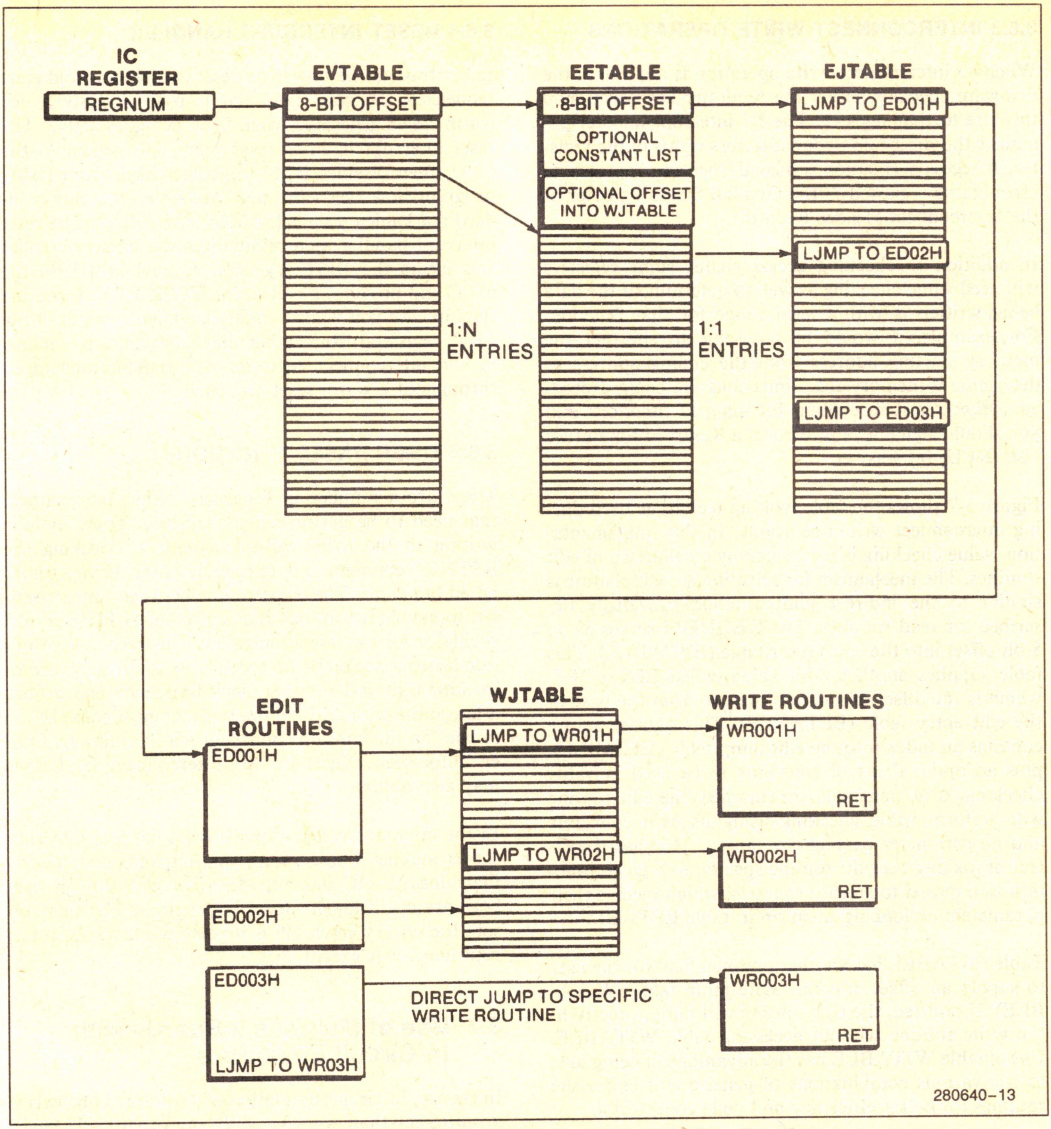


Figure 3-7. Indirect Jump Tables for Edit and Write Routines

3.6.3 INTERCONNECT WRITE OPERATIONS

When an interconnect write operation is requested, the firmware checks whether the requester has permission to write to that register. The IC interrupt handler examines the static and dynamic access rights of the register. If access permission is denied, then the appropriate error status is posted in the General Status Register of the Interconnect Header Record.

In addition to checking access rights, some registers will need value checking as well to determine if the data being written is within legal range for that function. For example, it would be advisable to subtract the memory starting address from the ending address to determine if memory size limitations are being exceeded. Other registers contain bit mapped functions and you should signal a value error if a Reserved for Future Use (RFU) bit was set.

Figure 3-7 shows the table look up technique used during interconnect write operations. In this implementation, value checking is performed by a collection of edit routines. The mechanism for selecting an edit routine is similar to the indirect jump method previously described for read routines. The REGNUM serves as an 8-bit offset into the edit vector table (EVTABLE). This table supplies another 8-bit value called EET_OFF, which is the offset of a data structure from the base of the edit entry table (EETABLE). This data structure contains an index into the edit jump table (EJTABLE), plus an optional list of constants to be used in value checking. If no constants are supplied, the edit routine will perform value checking using its own constants and no parameter passing is required. This has the effect of making the edit routine specific to a given function as opposed to generic routines, which receive their parameters by looking them up in table EETABLE.

Table EETABLE has another optional field for the user to supply an offset into the write jump table (WJTABLE). If omitted, the edit routine will jump directly to the write routine without accessing table WJTABLE. Use of table WJTABLE has the advantage of being able to use various combinations of generic edit and write routines for better efficiency and code compaction.

The indirect jump method of read, write, and edit services requires slightly more code space and software overhead than direct calling of register service routines, but offers a simple and extensible mechanism for users to form their own interconnect templates without needing to know the details contained within the core interconnect module. Instructions on how to load these tables and generate the final object code are presented in Section 5.

3.6.4 RESET INTERRUPT HANDLER

In contrast to the power-on reset (which is a cold start sequence), the reset interrupt is used exclusively for warm resets generated when UCINT* goes active. The reset interrupt handler asserts the reset signal to the CPU; then disables NMI to protect itself from losing program flow (yes—the non-maskable interrupt is indeed maskable on the 8751 Microcontroller!). The reset interrupt handler then reinitializes the microcontroller and clears the BIST registers. It next initializes the MPC and provides a call to the USER RESET routine (just in case you want to alter the default values in the MPC registers). Remember that any values previously in the interconnect registers will remain unchanged throughout a warm reset sequence.

3.6.5 TIMER BASED FUNCTIONS

There are a number of functions within interconnect that need to be performed periodically. These include writing to the MPC fail-safe counters, checking the RSTNC* counter, and optionally, user-defined timer based functions. The elapsed time between timer/counter interrupts is 1 ms based on use of an 11 MHz crystal oscillator for the microcontroller. This results in a fail-safe timeout period of 1.5 seconds. It is strongly recommended that you use this clock frequency and do not attempt the period of the timer interrupt because of the impact on the fail-safe count and RSTNC timing. Other unforeseen effects on the interconnect operations may also occur.

In the microcontroller, timer 0 is an internal interrupt source and has a higher hardware-implemented priority than does the IC interrupt handler even though they both are operationally level 1 interrupts. For this reason, the timer 0 interrupt is masked while the IC interrupt handler is executing.

3.7 Use of Multibus II Boards with In-Circuit Emulators

In theory, in-circuit emulation of Multibus II boards is no different than working with any other CPU based design. However, due to the indirect reset technique used within the interconnect subsystem, problems with reset initialization of the emulator probe are not uncommon. Most of these can be easily resolved by having the emulator issue port I/O commands to clear the RSTNC* bit in interconnect before a timeout occurs (typically 25 seconds). In some cases even this action will be inadequate because many ICE™ products will not tolerate a condition where reset is held continuously active. The solution to this problem is to disable the RSTNC* timeout function in the microcontroller. The core firmware design includes a jumper selectable option to allow in-circuit emulators to operate without reset problems.

3.8 Avoiding iPSB Timeouts

The reader will by now appreciate that interconnect bus cycles can be potentially long due to the extensive amount of processing a microcontroller completes to satisfy an interconnect reference. Performance within the interconnect address space is not an issue since interconnect bus cycles are run during system initialization and diagnostic tests, but cycles are rarely run during normal system operation because of their affect on the bus bandwidth. If, for any reason an interconnect bus cycle on the iPSB cannot complete within 1 ms; then an iPSB timeout will cancel the transaction and an error status posted. Therefore, users are cautioned not to attempt extensive processing within their interconnect read, write, and edit routines. Worst case timing analysis should be run on all new interconnect function records to avoid unexpected timeouts. More information on this topic is provided along with examples in Section 5.

4.0 INTERCONNECT HARDWARE DESIGN

4.1 Microcontroller Pin Assignments

The 8751 Microcontroller is well configured with I/O port facilities. In addition to the bidirectional Interconnect Address/Data bus (IAD<7..0>) connected to Port 0 (P0), there are 24 additional bidirectional lines and a number of port expansion techniques that can be employed. The core interconnect design will consume some of these resources, but beyond that, the user is free to dedicate the remaining I/O lines in any desired way. A variety of hardware design examples are presented here to satisfy the more typical requirements.

4.1.1 THE INTERCONNECT ADDRESS/DATA BUS

The initial design of the Message Passing Coprocessor (MPC) anticipated that the interconnect subsystem would be partitioned in a microcontroller of the 8751 or 8749 family. For this reason, the MPC provides a directly compatible multiplexed IAD bus requiring no external logic to the microcontroller. From a hardware viewpoint, the MPC is a slave to the microcontroller. The MPC supplies an external interrupt (**IREQ***) whenever it needs service and the microcontroller satisfies the request. The eight multiplexed IAD lines are taken directly from P0 of the microcontroller and interface timing is based on the microcontroller's Interconnect address strobe (**IAST**), Read (**IRD***), and Write (**IWR***) control signals. While the primary purpose of the IAD bus is to provide the communications path between the MPC and interconnect subsystem, it should be noted that the MPC will remain tristate for addresses 0 through 7FH. This condition presents an opportunity for I/O port expansion and is discussed later in this chapter.

4.1.2 RESERVED PINS ON THE 8751

In addition to the IAD bus on P0, the eight lines of Port 3 (P3) and two lines on Port 2 (P2) are reserved in support of the core interconnect design. (P3 is completely dedicated to hardware functions.) Figure 4-1 is a simplified schematic of the basic hardware requirements that guarantee proper functioning of the interconnect core. This includes the front panel LED, reset inputs and outputs, interrupt sources, and provisions for MPC loopback testing (discussed in Section 3).

Port function assignments are based on a series of equate statements contained in the program file DFT.MOD. If your interconnect design maintains the configuration shown in Figure 4-1 and defined in DFT.MOD; then there will be no need to modify the equate statements. A total of 14 port pins are available for user-defined I/O schemes. If necessary, the microcontroller pin assignments can be changed by modifying DFT.MOD (contained in the \ICFW\SRC subdirectory).

4.2 External TTL Options

Having completed the hardware design requirements, it is time to include the user hardware support options you have decided on. If your board requires only the core functions, then the hardware design is complete and prototyping can begin. Continued reading of this chapter may give you additional ideas for features you may wish to add. Other readers will already have a well defined functional objective and should be able to satisfy their requirements using one or more of the techniques given below. Bear in mind that the application examples described in this chapter are only suggestions and a combination of techniques are used. This underscores the need for good engineering discipline—always perform the AC/DC timing analysis and loading calculations!

The range of possibilities is limited only by your own imagination, but practical considerations will prevail. Keep aware of board real estate and bus loading requirements, expected implementation costs, and ask yourself if the feature is really worth the effort. With these considerations in mind, we will now examine some of the many options available for hardware design.

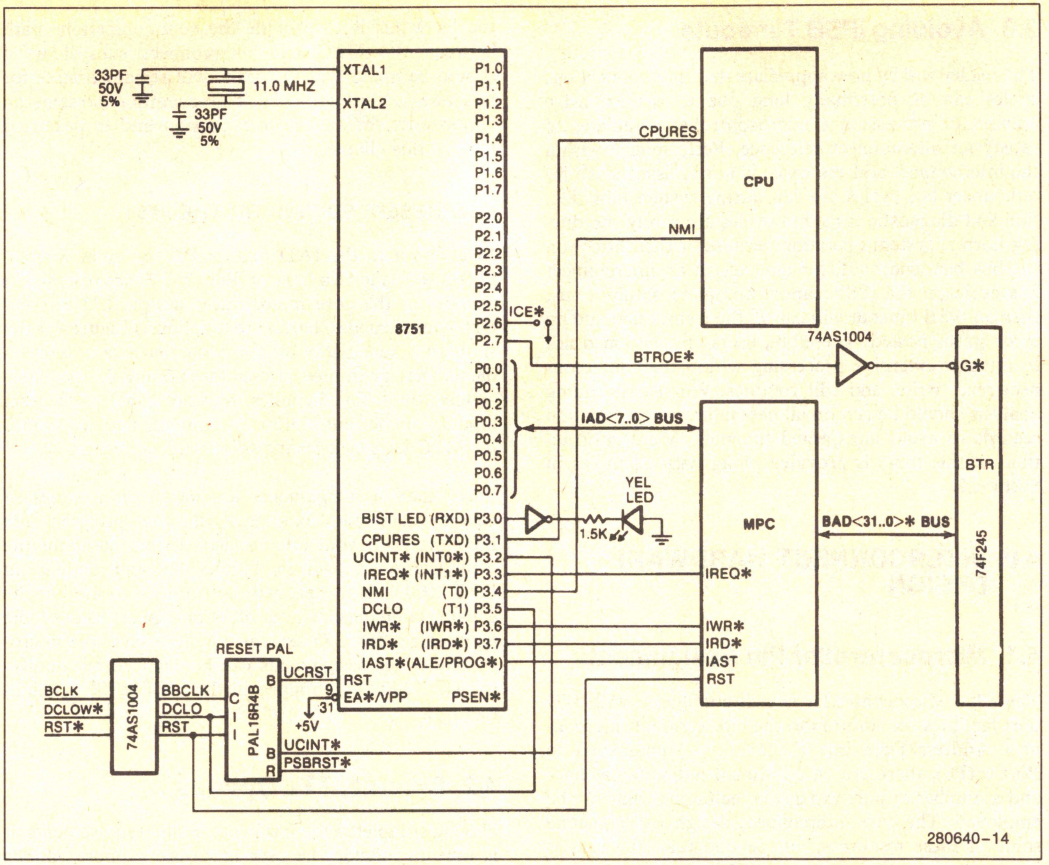


Figure 4-1. Interconnect Core Hardware Design

4.2.1 MICROCONTROLLER INPUT OPTIONS

On boards that contain user-supplied optional devices, it is desirable to include an interconnect status register to report whether or not the device is present. If present, system software can program the device with the appropriate driver. Some common examples include the Single Board Extension Bus (iSBX) MULTIMODULE compatible board products, numeric processors, and DMA controllers.

A technique used to detect a board's presence is to identify a port pin that ties to ground and then have the microcontroller read that pin. The P1 and P2 lines of the 8751 Microcontroller are internally pulled up, which makes module not present for these lines always read a logical "1". Verifying if a chip is present is more difficult since the component needs the ground pin for its own power consumption. The preferred technique here is to ask the user to install a jumper and in that way, the microcontroller can report correct status.

Figure 4-2 schematically diagrams various techniques used to input to an 8751 Microcontroller. A typical technique is to require the microcontroller to read jumper inputs. (Although Multibus II has reduced the number of jumpers required, it hasn't eliminated them.) In cases where jumpers are unavoidable (such as chip select jumpers and component present indicators), it would be helpful to report the state of those jumpers with an interconnect status register. An example is the EPROM size register found on many Intel boards. If you have only a small number of jumpers, then a direct connection to one of the microcontroller's port pins will suffice. For boards with a larger number of jumpers, buffering will be necessary. The buffer output enable term is generated by a port pin in association with the microcontroller's **IRD*** signal.

In the example shown, a read instruction to P2.5 will input data from the jumpers. This could be coded in ASM-51 as:

```
clrb      P2.5   ;activate CS for jumper
                 ;input buffer
mov   R0, 00h    ;load dummy address
movx  A   @R0,   ;read jumper inputs
setb      P2.5   ;deselect buffer
```

4.2.2 MICROCONTROLLER OUTPUT OPTIONS

Outputs from the microcontroller are used at various control points throughout Multibus II boards. Simple functions such as the LED's can use a direct connection to a microcontroller port pin. More complex functions will require a connection to the buffered IAD bus. As a general rule, whenever more than two loads are on the IAD bus, it will need buffering. The MPC must be connected directly to the IAD bus to ensure proper timing. The schematic in Figure 4-3 diagrams some typical applications.

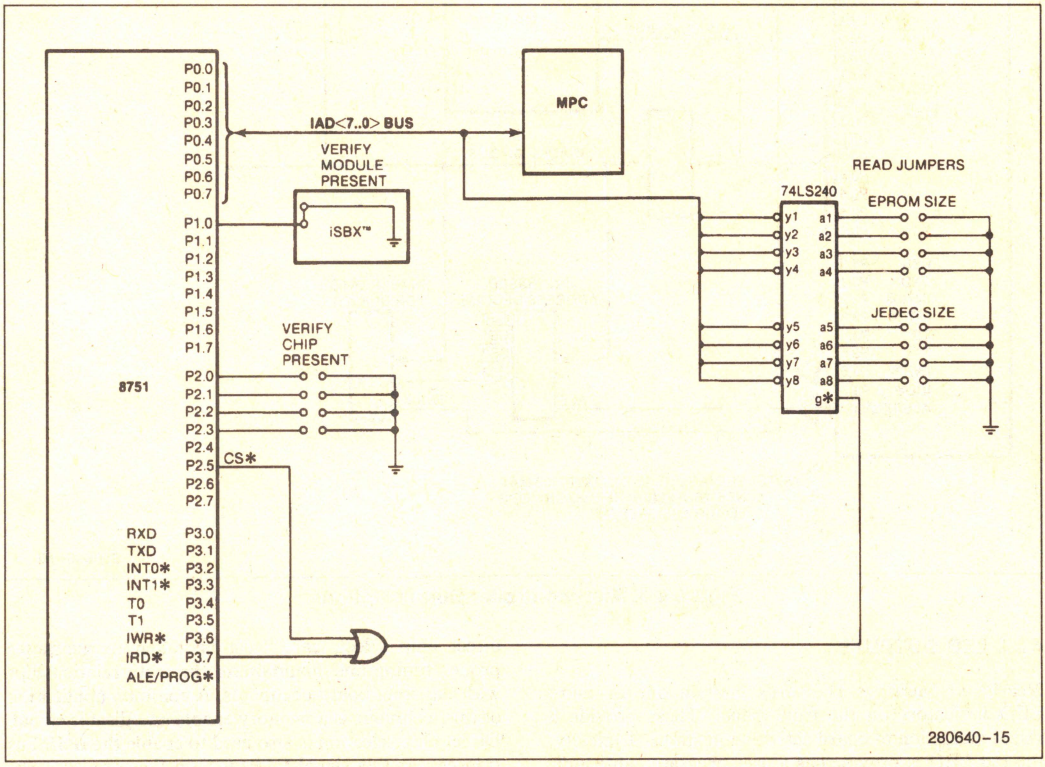


Figure 4-2. Microcontroller Input Options

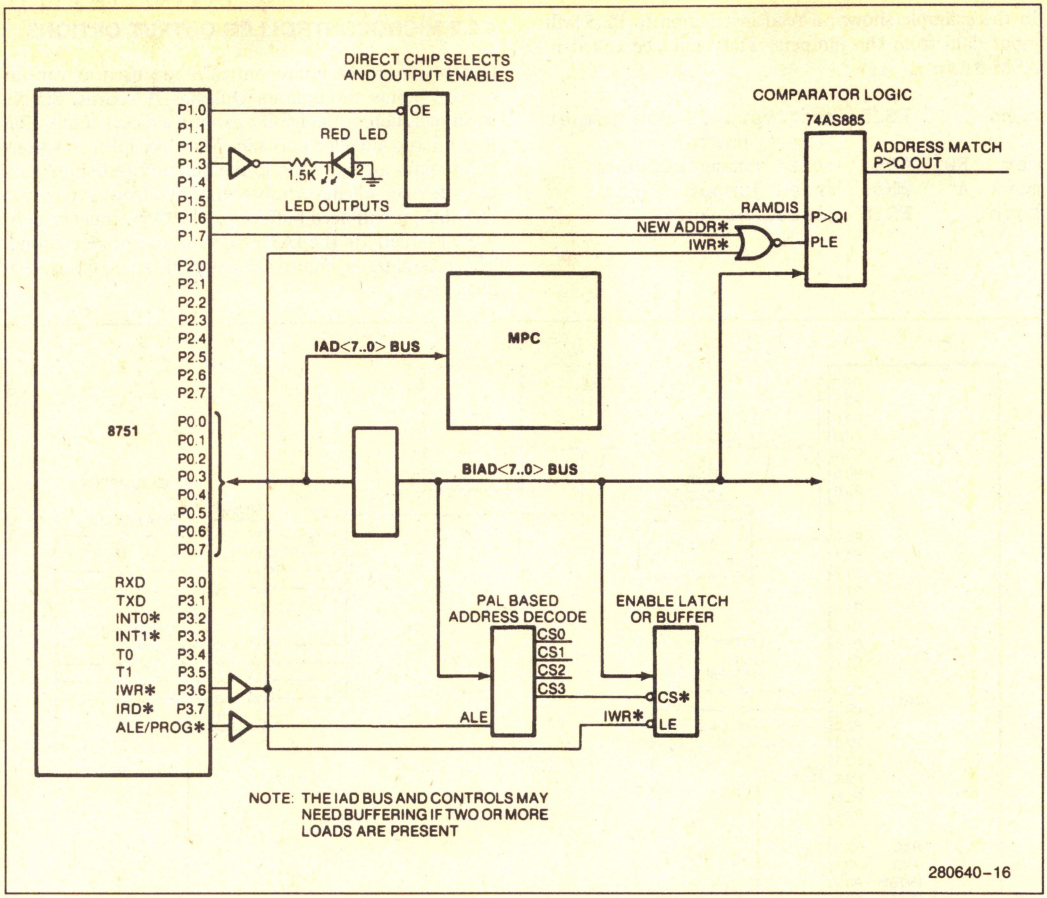


Figure 4-3. Microcontroller Output Options

4.2.3 LED OUTPUTS

Nearly all Multibus II boards contain one or more LED indicators on the front panel. These provide a visual indication of board activity and status. Typically, the red LED is provided as a user programmable indicator and is illuminated by the setting or clearing of a bit in a control register. A green LED is often used to indicate CPU activity. If the green LED is present, drive it with an Address Latch Enable (**ALE**) signal or equivalent. The yellow LED, if present, is lit during diagnostic testing and represents the ORed condition of the following bits in the BIST Slave Status Register:

BIST running + BIST failed + RSTNC timeout.

4.2.4 OUTPUT ENABLES AND CHIP SELECTS

A microcontroller port pin can be used for a bus transceiver, **ALE**, or a direct chip enable. When used as a direct chip select, care must be taken to guarantee proper timing and maintaining the state relationships with the other board occupants. A common application of this technique is a memory enable and disable signal. Direct chip selection is also used to enable the iPSB bus transceivers (via signal **BTROE***) in the core interconnect design.

4.2.5 ADDRESS DECODE ON THE IAD BUS

The IAD bus address can be mapped to generate chip selects using simple "1 of n" decoders gated by a **ALE** signal. As an alternative, a PAL-based decoder can be used. A variety of chip select signals can be generated using this technique. Some designs use the chip selects to enable secondary latches and buffers.

The ASM-51 coding for the circuit shown in Figure 4-3 could look something like this:

```
mov A,   ICDATA      ;Get value being written
mov R0,  Latch_addr  ;Load the address of the
                     ;latch
movx@R0,A            ;Generate chip select and
                     ;write data
```

4.2.6 COMPARATORS

Comparators are often used for address boundary checks in association with Local Bus Extension (iLBX) or local memory. A comparator logic example is shown in Figure 4-3. A select term can be generated on greater than, less than, or equal to the data placed on the IAD bus. The set point for the comparison is latched-in during the data phase of the microcontroller's P0 bus. In this way, a port pin (the **NEW ADDR*** signal) can combine with the microcontroller's **IWR*** to control when the address comparison is enabled. The ASM-51 code to load a new address into the comparator could look like this:

```
mov  A,    ICDATA    ;Get the new setpoint
clrb       NEW_ADDR  ;This is the PLE signal
mov  R0,   00h       ;Dummy address
movx @R0,  A         ;Output, the new address
setb       NEW_ADDR  ;Latch it in
```

4.3 Bidirectional I/O on the IAD Bus

Even a modest implementation of interconnect can easily exceed the available port resources of the microcontroller, especially when a byte wide data path is required. For these applications, numerous port expansion techniques exist. Your choice of which technique is best in your particular design will be based on such factors as on board space requirements, cost, bidirectionality, and ease of programming. An example of interfacing a complex peripheral to the IAD bus is shown in Figure 4-4. The ASM-51 coding for this circuit is very straightforward since device selection is based on an IAD bus address.

```
mov R0,     IO_ADDR ;This is the address of
                    ;the device
movxA,      @R0     ;Input from the device
mov ICDATA,A        ;Save the data

mov R0,     IO_ADDR ;This is the address of
                    ;the device
mov A,      ICCNTR  ;Load a control value for
                    ;output
movx@R0,    A       ;Output to the device
```

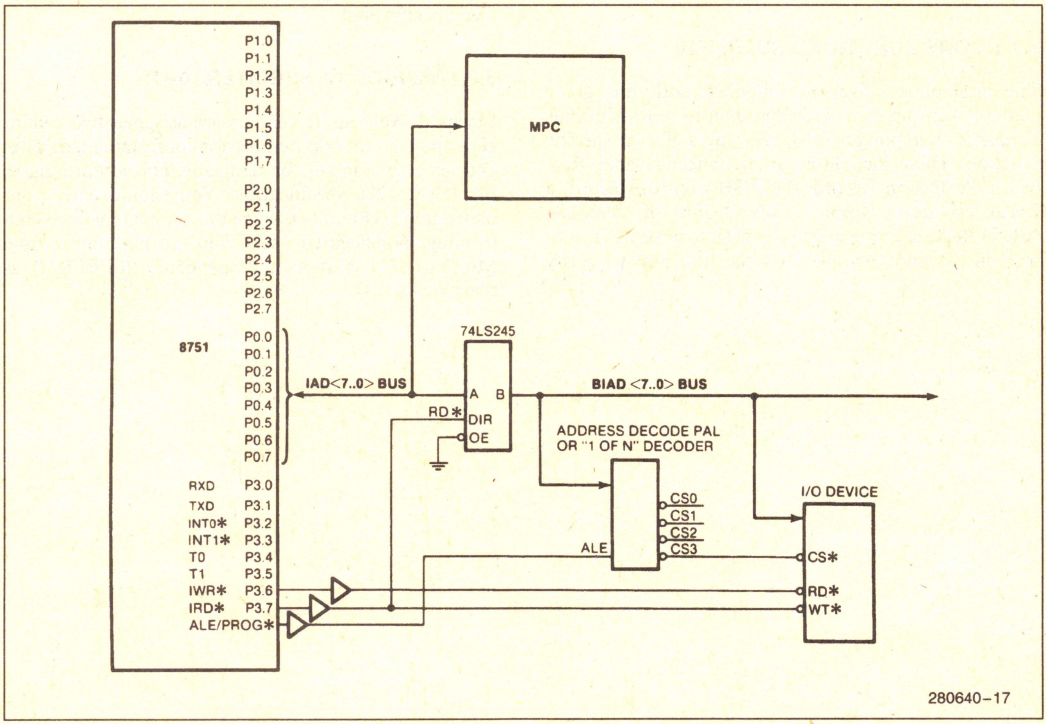


Figure 4-4. Bidirectional I/O on the IAD Bus

5.0 DEFINING THE INTERCONNECT TEMPLATE

5.1 What Goes into Interconnect?

The design implementation phase of your interconnect subsystem begins with a careful evaluation of what functions to include. Common core functions are supplied on the accompanying diskette and you can add supplemental records to the interconnect template for board-level features you intend to support. Many frequently used function records are already defined in the Interconnect Interface Specification and the reader is requested to review this document before proceeding. If none of the furnished function records meet your needs, you can create your own function record using one of the record types available for vendor definition or board specific functions.

If your function record is likely to have broad industry appeal and could be used by other vendors to support similar functions, you may wish to fill out the petition application included with the reader comment sheet at rear of this guide. Your petition will be evaluated based on suitability for a general class of hardware, industry standardization, and feedback from the Multibus II user community, and if accepted, published in the next revision to Interconnect Interface Specification.

5.1.1 CORE FUNCTIONS SUPPLIED

The interconnect firmware supplied with this guide contains a group of records common to most Multibus II boards containing a CPU, regardless of their specific function. These include an Interconnect Header Record, a Protection Record, the iPSB Control Record, a Local Processor Record, and an End of Template (EOT) Record. Some users may wish to generate a minimal interconnect template on the first pass to verify the operation of their base har are before adding their own advanced functions. This approach simplifies debugging and allows new features to be added incrementally.

5.1.2 GETTING STARTED: THE INTERCONNECT WORKSHEET

Unless you are using the supplied interconnect template without modification, it is a two-step process to complete the configuration tables. In the first step, you will fill-in a worksheet with information about default values, RAM usage, read/write and edit routine numbers, and other details. Figure 5-1 is an example of such a worksheet. While completing these tables, you might notice opportunities to reduce code size by making use of generic read, write, and edit routines. In the second step, the information in the worksheet is copied into a series of tables used by the configuration macros.

5.2 The Interconnect Header Record

The interconnect template begins with a standard Interconnect Header Record. This record contains fields that require specific values supplied by the user. You should fill-in this information in as we proceed through this chapter. For a detailed discussion concerning each register, consult Appendix A of the Interconnect Interface Specification.

5.2.1 VENDOR ID REGISTER (0–1)

Licensed Multibus II vendors are assigned two vendor ID numbers—an odd number for conforming templates and an even number for templates not conforming to the IEEE 1296 specification. You should enter your odd vendor ID number, since this template will be conforming. Nonlicensed users who are building custom Multibus II boards should use 65533 (0FFFDH) as their vendor ID.

REGISTER NAME	GAR	RAM	DFT	RRT	ERT	WRT	CONST	EET	VAR
HEADER RECORD									
VENDOR ID, LOW BYTE	0	0	____	000H	000H	?	?	000H	?
VENDOR ID, HIGH BYTE	0	0	____	000H	000H	?	?	000H	?
BOARD ID, CHARACTER 1	0	0	____	000H	000H	?	?	000H	?
BOARD ID, CHARACTER 2	0	0	____	000H	000H	?	?	000H	?
BOARD ID, CHARACTER 3	0	0	____	000H	000H	?	?	000H	?
BOARD ID, CHARACTER 4	0	0	____	000H	000H	?	?	000H	?
BOARD ID, CHARACTER 5	0	0	____	000H	000H	?	?	000H	?
BOARD ID, CHARACTER 6	0	0	____	000H	000H	?	?	000H	?
BOARD ID, CHARACTER 7	0	0	____	000H	000H	?	?	000H	?
BOARD ID, CHARACTER 8	0	0	____	000H	000H	?	?	000H	?
BOARD ID, CHARACTER 9	0	0	____	000H	000H	?	?	000H	?
BOARD ID, CHARACTER 10	0	0	____	000H	000H	?	?	000H	?
INTEL RESERVED	0	0	000H	000H	000H	?	?	000H	?
INTEL RESERVED	0	0	000H	000H	000H	?	?	000H	?
INTEL RESERVED	0	0	000H	000H	000H	?	?	000H	?
INTEL RESERVED	0	0	000H	000H	000H	?	?	000H	?
HARDWARE TEST REV NO.	0	0	____	000H	000H	?	?	000H	?
CLASS ID	0	0	____	000H	000H	?	?	000H	?
RFU	0	0	000H	000H	000H	?	?	000H	?
RFU	0	0	000H	000H	000H	?	?	000H	?
RFU	0	0	000H	000H	000H	?	?	000H	?
RESET STATUS RGTR	0	1	003H	001H	000H	?	?	000H	RST
PROGRAM TABLE INDEX	1	1	000H	001H	001H	000H	00H	001H	PTI
NMI ENABLE RGTR	1	1	000H	001H	001H	000H	38H	002H	NMI_EN
GENERAL STATUS	0	1	000H	003H	000H	?	?	000H	GEN_STS
GENERAL CONTROL	1	1	000H	001H	001H	002H	78H	003H	GEN_CTL
BIST-SUPPORT-LEVEL	0	1	000H	001H	005H	000H	18H,07H,04H	004H	BIST_SL
BIST-DATA-IN	1	1	000H	001H	001H	000H	00H	001H	BIST_DI
BIST-DATA-OUT	0	1	000H	001H	001H	000H	00H	001H	BIST_DO
BIST-SLAVE-STATUS	0	1	010H	001H	001H	000H	00H	001H	BIST_SS
BIST-MASTER-STATUS	1	1	020H	001H	001H	000H	50H	005H	BIST_MS
BIST-TEST-ID	0	1	000H	001H	002H	000H	01H,0FEH	006H	BIST_TID
PROTECTION RECORD									
PROTECTION RECORD TYPE	0	0	00BH	000H	000H	?	?	000H	PROT_OFF
RECORD LENGTH	0	0	002H	000H	000H	?	?	000H	?
PROTECTION LEVEL RGTR	0	2	000H	004H	001H	003H	0FEH	007H	UACRS
RFU	0	0	000H	000H	000H	?	?	000H	?
PSB CONTROL RECORD									
PSB CONTROL RECORD TYPE	0	0	006H	000H	000H	?	?	000H	PSB_CTL_OFF
RECORD LENGTH	0	0	006H	000H	000H	?	?	000H	?
PSB SLOT ID NO.	0	0	000H	002H	000H	?	?	000H	?
PSB ARBITRATION ID NO.	1	0	000H	002H	003H	?	?	008H	?
PSB ERROR RGTR	1	0	000H	002H	001H	001H	0FFH	009H	?
PSB CONTROL/STATUS RGTR	1	1	01FH	005H	001H	004H	63H	00AH	PSB_CS
PSB DIAGNOSTICS RGTR	1	1	000H	002H	004H	?	?	00BH	PSB_DIAG
RFU	0	0	000H	000H	000H	?	?	000H	?

280640-18

Figure 5-1. Interconnect Configuration Worksheet (Sheet 1 of 2)

5

REGISTER NAME	GAR	RAM	DFT	RRT	ERT	WRT	CONST	EET	VAR
----------LOCAL PROC RECORD ----------									
LOCAL PROC RECORD TYPE	0	0	013H	000H	000H	?	?	000H	?
RECORD LENGTH	0	0	003H	000H	000H	?	?	000H	?
LOCAL PROC CONTROL	0	0	000H	006H	001H	006H	0FEH	00CH	?
LOCAL PROC STATUS	0	0	000H	007H	000H	?	?	000H	?
RFU	0	0	000H	000H	000H	?	?	000H	?
---------- END OF TEMPLATE RECORD ----------									
EOT RECORD TYPE	0	0	0FFH	000H	000H	?	?	000H	EOT_OFF

280640-19

Figure 5-1. Interconnect Configuration Worksheet (Sheet 2 of 2)

5.2.2 BOARD ID REGISTERS (2-11)

These 10 registers contain the ASCII product code as described in your user's manual. You should avoid the use of nonprinting ASCII characters so the system software can display the product code verbatim on a CRT screen. If the product code is less than 10 characters, pad the remaining register fields with ASCII null characters (zeros).

5.2.3 HARDWARE TEST REVISION NUMBER REGISTER (16)

This register is part of a system of control for matching the firmware revision to the automatic test equipment used with that board. The master or local test handler software uses this as a mechanism for revision control of diagnostics.

5.2.4 CLASS ID REGISTER (17)

Determine your calss ID from the list supplied in the Interconnect Interface Specification. The class ID occupies the most significant nibble (4 bits) of the register and the subclass ID occupies the least significant nibble.

5.2.5 BIST SUPPORT LEVEL REGISTER (26)

The BIST Support Level Register defines the support level of the diagnostics on-board. This will determine which commands can be issued to your board via the BIST Data Input Register.

5.3 OPTIONAL USER FUNCTIONS

There is virtually no limit to the number and variety of function records that users can add to the core firmware—provided the following common sense guidelines are met:

a. A maximum of 256 interconnect registers are allowed due to the page limit length of the configuration tables. Most boards will need fewer than 100 registers.

b. The microcontroller you are using has certain resource limitations—the most pressing of which is on-chip RAM. Nested stack operations also require RAM, so you must avoid recursive procedures and nesting of calls beyond four levels in any of the USER code modules. Normally, it is preferable to pass parameters in registers rather than on the stack.

c. An interconnect request must complete operations within 1 ms to avoid an iPSB timeout. The actual time available to complete a USER routine is less than 0.5 ms because interconnect is dual ported and a local access can temporarily lockout an iPSB request. A timing analysis of your design will be done prior to PROMming the code (refer to paragraph 5.8.2). (Sheet 1 of 2)

d. Caution must be exercised whenever the potential for deadlock of a resource exists. Solutions to this problem include prioritization, reporting of error status, or timeout to guarantee that system hand does not occur.

e. User read, write and edit routines must follow the naming convention described in this chapter to guarantee that the macro expansion works properly. Any global variable names used within existing core modules are considered reserved and should not be used in your own procedures (ASM-51 and/or RL51 will report duplicate symbol errors). A list of existing public variables may be found in program file ICFW\SRC\USER.M (Sheet 2 of 2)OD.

5.4 Completing the Interconnect Configuration Worksheet

If you decided that a user-defined function record is to be incorporated in the interconnect template, then you must define the access rights, default values, and com-

plete the remaining worksheet entries needed by the microcontroller firmware to support those features. You then write the functional read, write, and edit routines and tag them with appropriate labels. The new function record(s) must be inserted ahead of the EOT record of the core interconnect template.

As discussed in Chapter 3, the interconnect firmware uses a series of tables to associate a register with the correct read, write, and edit routine numbers. The information you supply here will be loaded into those tables during the code assembly process. In Figure 5-2, an example of a completed interconnect configuration worksheet for the core firmware is shown. The parameters that must be supplied for each interconnect register identified on the configuration worksheet are described below.

5.4.1 GLOBAL ACCESS RIGHTS

All interconnect registers are readable from both the local and the iPSB bus. The state of the Global Access Rights (GAR) bit determines whether the register can be written to by another iPSB agent. If this bit is 0, then the register is read only on the iPSB bus and edit routine ED00H is used. A register will be locally read/writable only if an edit routine other than ED00H is supplied for this entry. If the GAR bit is 1, then this register is read/writable on both interfaces and an edit routine must be supplied.

5.4.2 RAM USAGE

The microcontroller RAM is a precious resource and must be managed as efficiently as possible. Some registers in interconnect may contain static values that never change. These values should be stored in microcontroller ROM. In cases where the contents of an inteconnect register are variable, either a byte or a bit of RAM must be allocated to store data, unless the data can be read directly from a port. The microcontroller's bit manipulation features can be used when a full 8-bit register is not required. The RAM entry in the worksheet must specify whether to reserve a bit location (RAM = 2), a byte location (RAM = 1), or no memory at all (RAM = 0). If RAM memory is requested, then a name for the public variable for that location must be supplied (refer to paragraph 5.4.9).

5.4.3 DEFAULT VALUE

The Default Value (DFT) is a hex number placed in this register during a cold or recovery reset. If this is a static register, then the default value represents the permanent contents of that register. The default value is also placed into dynamic registers following a cold reset; however, the register contents may subsequently change due to reprogramming or environmental changes (e.g., reading a value from a port).

5.4.4 READ ROUTINE ENTRY

Read routines contain the code that actually perform the work in satisfying an interconnect read request. Eight read routines are used in the interconnect core firmware. These are listed below and all are user callable. Read routines are sequentially numbered from 00H to nnH and contain no numbering gaps. A single read routine may be referenced by several different interconnect registers. If you are adding a new read routine, the first available Read Routine (RRT) number would be RD08H. The value supplied for the RRT is the two digit hex portion of the read routine number.

RD00h	Load Default Value into ICDATA	Generic
RD01h	Get RAM Access Rights	Generic
RD02h	PSB Control/Status Register	Specific
RD03h	General Status Register	Specific
RD04h	Protection Level Register	Specific
RD05h	MPC General Parameter Register	Specific
RD06h	Local Processor Control Register	Specific
RD07h	Local Processor Status Register	Specific

```
;*******************************************************
;
;  NAME:          TABGEN  (TABLE GENERATION)
;
;  DESCRIPTION:   USES ENTAB MACRO TO ASSOCIATE EACH IC
;                 REGISTER WITH:
;                       GAR - GLOBAL ACCESS RIGHTS
;                       RAM - RAM USEAGE (WHETHER OR
;                             NOT UC RAM IS NEEDED)
;                       DFT - DEFAULT VALUE
;                       RRT - READ ROUINTE NUMBER
;                       EET - EDIT ENTRY NUMBER
;                       VAR - SYMBOL FOR RAM/BIT
;                             VARIABLE
;
; UPDATE HISTORY: 5-14-87 JR UPDATE FOR IDX
;
;*******************************************************
;       G R
;       A A
;       R M  DFT  RRT  EET  VAR
;-------------- HEADER RECORD -------------------------
%ENTAB(0,0,001H,000H,000H,?)       ;VENDOR ID, LOW BYTE
%ENTAB(0,0,000H,000H,000H,?)      ;VENDOR ID, HIGH BYTE
%ENTAB(0,0,000H,000H,000H,?)     ;BOARD ID, CHARACTER 1
%ENTAB(0,0,000H,000H,000H,?)     ;BOARD ID, CHARACTER 2
%ENTAB(0,0,000H,000H,000H,?)     ;BOARD ID, CHARACTER 3
%ENTAB(0,0,000H,000H,000H,?)     ;BOARD ID, CHARACTER 4
%ENTAB(0,0,000H,000H,000H,?)     ;BOARD ID, CHARACTER 5
%ENTAB(0,0,000H,000H,000H,?)     ;BOARD ID, CHARACTER 6
%ENTAB(0,0,000H,000H,000H,?)     ;BOARD ID, CHARACTER 7
%ENTAB(0,0,000H,000H,000H,?)     ;BOARD ID, CHARACTER 8
%ENTAB(0,0,000H,000H,000H,?)     ;BOARD ID, CHARACTER 9
%ENTAB(0,0,000H,000H,000H,?)    ;BOARD ID, CHARACTER 10
%ENTAB(0,0,000H,000H,000H,?)            ;INTEL RESERVED
%ENTAB(0,0,000H,000H,000H,?)            ;INTEL RESERVED
%ENTAB(0,0,000H,000H,000H,?)            ;INTEL RESERVED
%ENTAB(0,0,000H,000H,000H,?)            ;INTEL RESERVED
%ENTAB(0,0,001H,000H,000H,?)    ;HARDWARE TEST REV NO.
%ENTAB(0,0,000H,000H,000H,?)                  ;CLASS ID
%ENTAB(0,0,000H,000H,000H,?)                      ;RFU
%ENTAB(0,0,000H,000H,000H,?)                      ;RFU
%ENTAB(0,0,000H,000H,000H,?)                      ;RFU
;       G R  DFT  RRT  EET  VAR
%ENTAB(0,1,003H,001H,000H,RST_STS)  ;RESET STATUS RGTR
%ENTAB(1,1,000H,001H,001H,PTI)     ;PROGRAM TABLE INDEX
%ENTAB(1,1,000H,001H,002H,NMI_EN)      ;NMI ENABLE RGTR
%ENTAB(0,1,000H,003H,000H,GEN_STS)     ;GENERAL STATUS
%ENTAB(1,1,000H,001H,003H,GEN_CTL)     ;GENERAL CONTROL
%ENTAB(0,1,000H,001H,004H,BIST_SL) ;BIST-SUPPORT-LEVEL
%ENTAB(1,1,000H,001H,001H,BIST_DI)        ;BIST-DATA-IN
                                                280640-20
```

Figure 5-2. ENTAB Macro Source Code Listing (Sheet 1 of 2)

```
%ENTAB(0,1,000H,001H,001H,BIST_DO)       ;BIST-DATA-OUT
%ENTAB(0,1,010H,001H,001H,BIST_SS)  ;BIST-SLAVE-STATUS
%ENTAB(1,1,020H,001H,005H,BIST_MS) ;BIST-MASTER-STATUS
%ENTAB(0,1,000H,001H,006H,BIST_TID)       ;BIST-TEST-ID
;------------ PROTECTION RECORD ----------------------
;       G R  DFT  RRT  EET  VAR
%ENTAB(0,0,008H,000H,000H,PROT_OFF)  ;PROTECT REC TYPE
%ENTAB(0,0,002H,000H,000H,?)             ;RECORD LENGTH
%ENTAB(0,2,000H,004H,007H,UACRS)    ;PROTECT LEVEL RGTR
%ENTAB(0,0,000H,000H,000H,?)                      ;RFU
;------------ PSB CONTROL RECORD ---------------------
;       G R  DFT  RRT  EET  VAR
%ENTAB(0,0,006H,000H,000H,PSB_CTL_OFF)     ;PSB CR TYPE
%ENTAB(0,0,006H,000H,000H,?)             ;RECORD LENGTH
%ENTAB(0,0,000H,002H,000H,?)           ;PSB SLOT ID NO.
%ENTAB(1,0,000H,002H,008H,?)           ;PSB ARB ID NO.
%ENTAB(1,0,000H,002H,009H,?)           ;PSB ERROR RGTR
%ENTAB(1,1,01FH,005H,00AH,PSB_CS)  ;PSB CONT/STAT RGTR
%ENTAB(1,1,000H,002H,00BH,PSB_DIAG)      ;PSB DIAG RGTR
%ENTAB(0,0,000H,000H,000H,?)                      ;RFU
;------------ LOCAL PROC RECORD ----------------------
;       G R  DFT  RRT  EET  VAR
%ENTAB(0,0,013H,000H,000H,?)   ;LOCAL PROC RECORD TYPE
%ENTAB(0,0,003H,000H,000H,?)             ;RECORD LENGTH
%ENTAB(0,0,000H,006H,00CH,?)        ;LOCAL PROC CONTROL
%ENTAB(0,0,000H,007H,000H,?)         ;LOCAL PROC STATUS
%ENTAB(0,0,000H,000H,000H,?)                      ;RFU
;--------- END OF TEMPLATE RECORD --------------------
;       G R  DFT  RRT  EET  VAR
%ENTAB(0,0,0FFH,000H,000H,EOT_OFF)     ;EOT RECORD TYPE
$EJECT
                                                280640-21
```

Figure 5-2. ENTAB Macro Source Code Listing (Sheet 2 of 2) (Continued)

5.4.5 EDIT ROUTING TABLE

The purpose of the Edit Routine Table (ERT) parameter in the worksheet is to associate each register with the correct edit routine number. There are six edit routines supplied within this firmware:

ED00h	Used for Read-Only Registers	Generic
ED01h	Check that RFU Bits Match a Mask	Generic
ED02h	Range Checking Between Two Constants	Generic
ED03h	PSB ARB ID Register	Specific
ED04h	PSB Diagnostic Register	Specific
ED05h	Greater Than Check (>)	Generic

5.4.6 WRITE ROUTINE NUMBER

The Write Routine (WRT) parameter specifies which write routine number to jump to once access rights and value checking are complete. A given edit routine can jump to any write routine (there is no requirement for the edit routine number to match the write routine number). In the generic routines, several edit routines can reference the same write routine using different constant lists based on register number. In the specific routines, it is sometimes more expedient to jump directly to the edit routine rather than going through the write jump table. In this case, a "?" should be entered for the WRT. Seven write routines are supplied, therefore, the first available number assignment for user code is WR07H.

WR00h	Write to a RAM Based Register	Generic
WR01h	Write to a MPC Based Register	Generic
WR02h	Write to PSB Control Register	Specific
WR03h	Write to Protection Level Register	Specific
WR04h	Write to PSB Control/Status Register	Specific
WR05h	Write to PSB Diagnostic Control	Specific
WR06h	Write to Local Processor Control	Specific

5.4.7 CONSTANTS FOR VALUE CHECKING

This column contains a list of constants (CONST) used by the edit routines in value checking register contents. During an edit routine, illegal values cause error reports to the General Status Register and the IC handler will exit without performing the write operation. The advantage of using the CONST is that edit routines can be made "sharable" because a different list of constants is used for each register. If no constants are required, a "?" should be entered. This implies that your value checking algorithm will use constants supplied by the edit routine (i.e., specific to that register).

5.4.8 THE EDIT ENTRY TABLE

The Edit Entry Table (EET) number is a value used as the index into the edit entry table during table look up operations. The order of how EET numbers get assigned is not important, however, numbering must start at 0H and run in sequence until every register has an assigned EET number. Two or more registers having the same attributes share the same EET number. This is true if (and only if) all of the columns match for the ERT, WRT, and CONST values. When this occurs, there is an opportunity for code compaction and the same EET number will be assigned to two or more registers.

5.4.9 SYMBOLIC REFERENCE TO RAM LOCATIONS

Whenever a dynamic register is defined (RAM = 1 or 2), an on-chip RAM bit or memory location is reserved. Entries in the Symbolic Reference to RAM (VAR) column represent the symbolic name to be used for references to that RAM bit or memory location. The VAR entry serves an important purpose—it reserves a location in on-chip memory and declares a public symbol under the name of that variable. Read, write and edit routines can now reference that variable directly without needing to know its exact location.

One word of caution with respect to bit variable declarations. If the RAM parameter in the worksheet is a byte location (RAM = 1), then the value listed as default will be loaded into the location reserved for that symbolic name at initialization time as expected (VAR = DFT). But if a bit location was reserved (RAM = 2), the current software has no provision to load the reserved bit location with its default value. Users are therefore advised to initialize all reserved bit locations explicitly in their INIT_USER routines. The public variable specified by the VAR parameter will be in effect and can be used as a symbolic reference for direct addressing (e.g., setb/clrb var). An example of the use of bit variables can be found in the RD04 and WR03 routines contained in Appendix C.

5.5 Loading the Macro Tables

The interconnect worksheet was used as an intermediate step to allow the user to identify opportunities for code reduction through the use of generic read, write, and edit routines. Now that the worksheet is complete, you must enter this data into table generating macro files on your IBM PC. The macro assembler will use these tables to generate the final interconnect firmware code.

5.5.1 THE ENTAB TABLE

The program file \ICFW\SRC\TABGEN.A51 contains a list of calls to the ENTAB configuration macro (Figure 5-2). This list determines the register order in the interconnect template. Enter the values for GAR, RAM, DFT, RRT, EET, and VAR from the worksheet into this table.

5.5.2 THE EETGEN TABLE

The program file \ICFW\SCR\EETGEN.A51 contains a list of calls to the EETGEN configuration macro (Figure 5-3). This list is used to build the edit entry table. Enter the values for ERT, CONST, and WRT from the worksheet into this table.

```
;*******************************************************
;
;  NAME:          EELBL (EDIT ENTRY TABLE)
;
;  PURPOSE:       PROVIDE EDIT ROUTINE ADDRESSES,
;                 VALUE CHECKING CONSTANTS, AND WRITE
;                 ROUTINE ADDRESSES FOR THE IC MODULE
;
;  DESCRIPTION:   A ONE DIMENSIONAL LOOKUP TABLE THAT
;                 SPECIFIES LOW BYTES OF ADDRESSES
;                 THAT RESULT FROM EXPRESSIONS, i.e.
;                 (ERXX-ERLBL), AND ALSO PROVIDES
;                 CONSTANTS THRU THE USE OF SYMBOLS
;
;  UPDATE HISTORY: 5-14-87 JR UPDATED FOR IDX
;
;*******************************************************
;
;      ERT  CONST  WRT

%EEGEN(00H,%(?),?)              ;READ ONLY REGISTERS    0
%EEGEN(01H,%(00H),00H)          ;PROGRAM TABLE INDEX    1
%EEGEN(01H,%(038H),00H)         ;NMI ENABLE             2
%EEGEN(01H,%(078H),02H)         ;GENERAL CONTROL        3
%EEGEN(05H,%(018H,07H,04H),00H) ;BIST SUPPORT LVL       4
%EEGEN(01H,%(050H),00H)         ;BIST MASTER STATUS     5
%EEGEN(02H,%(01H,0FEH),00H)     ;BIST TEST ID           6
%EEGEN(01H,%(0FEH),03H)         ;PROTECTION LEVEL       7
%EEGEN(03H,%(?),?)              ;PSB ARB. ID            8
%EEGEN(01H,%(0FFH),01H)         ;PSB ERROR              9
%EEGEN(01H,%(063H),04H)         ;PSB CONTROL STATUS     A
%EEGEN(04H,%(?),?)              ;PSB DIAGNOSTIC         B
%EEGEN(01H,%(0FEH),06H)         ;LOCAL PROC CONTROL     C
$EJECT
```

280640-22

Figure 5-3. EEGEN Macro Source Code Listing

5.5.3 THE EXTERNS MACRO: GENERATING EXTERNAL LABELS

After completing the entries in the ENTAB and EETGEN tables, the final step before code generation is to supply values to the EXTERNS macro so that the correct number of external labels are generated for your user routines. The EXTERNS macro is used in the file \ICFW\SRC\TABLE.MOD. Here you will find the call to EXTGEN, where you must supply the highest number of your edit, write, and read routines as parameters in that order.

```
%EXTERNS(06,07,08)          ;Six edit routines
                            ;Seven write routines
                            ;Eight read routines
```

5.6 Programming Techniques for Read, Write, and Edit Routines

There are a group of user callable macro functions supplied with the interconnect firmware in the file \ICFW\SRC\GLOBAL.MAC. These are called by entering a "%" symbol in front of the name and supplying a list of parameters. This results in code expansion, which supplies the requested function. The following macros are callable by users.

```
RD_MPC (dest,src)    Read MPC Register
WRT_MPC(dest,src)    Write MPC Register
LOOKUP(table,offset)Retrieve parameter from
                     table
SUB (opr)            Subtract operand from
                     accumulator
MOVBIT
  (dest_bit,src_bit)Perform bit move
                     operation
GET_EEC              Get Edit Table Constant
                     (Data pointer must be
                     pointing to EETABLE.
                     Byte is returned in
                     Accumulator. You must
                     perform one LOOKUP be-
                     fore making this call.)
```

All user routines should be declared PUBLIC so that the table generating macros can locate the routines entry point. Place the user routines you write in the file ICFW\SRC\USER.MOD.

5.6.1 RETRIEVING THE CONSTANT LIST FROM THE EETABLE

While in a read, write, or edit routine, one or more constants can be retrieved by using the global symbol EET__OFF as the offset into the EETABLE for this register.

For example, consider this range checking algorithm in an edit routine:

```
ER05H:  INC   EET_OFF           ;Point to RFU Mask
                                ;in EETABLE
        %LOOKUP
         (#EETABLE,EET_OFF)     ;Get It
        ANL   A,ICDATA          ;Check all RFUs=0
        JNZ   ERR_5             ;JMP if illegal
        %GET_EEC                ;Get Don't Care Mask
                                ;From EETABLE
        ANL   A,ICDATA          ;Mask off Don't Care
                                ;Bits from ICDATA
        MOV   TEMP,A            ;Save Result
        %GET_EEC                ;Get Max Allowable
                                ;Write Value
        %SUB(TEMP)              ;SUB Value Being
                                ;Written
        JC    ERR_5             ;If ICDATA > Max
                                ;Allowable, ERR
        %GET_EEC                ;Else Get Write
                                ;Routine Offset
        MOV   DPTR,#WRLBL       ;Point to WR
        JMP   @A+DPTR           ;GOTO WR
ERR_5   AJMP  VALERR            ;Report Value ERR
```

5.6.2 HANDLING OF VALUE ERRORS

If a value error is detected in an edit routine, then the write operation will not be performed and instead, a jump to a public routine VALERR should be executed. This routine updates the General Status Register with the appropriate Value Error and returns to the IC handler to complete the interconnect operation.

5.7 Hooks for User Supplied Routines

Every board design is just a little bit different and recognizing this fact, calls to user supplied routines are provided at certain critical locations in the code. These include initialization, polling, and reset routines and timer based functions. As delivered, these routines are nothing more than program stubs. You can find them in the \ICFW\SRC\USER.MOD file. The following discussion suggests some ideas for what you might do with the user routine calls in your interconnect design.

5.7.1 INIT USER: CUSTOM INITIALIZATION CODE

The INIT_USER routine is only called on a cold reset. It gives the user an opportunity to initialize their I/O and modify defaults before entering the mainline code. This routine also gives the user a chance to modify the MPC register defaults after the call to MPC_INIT. If you are using any bit segments (RAM = 2), they must be explicitly initialized at this time. A special feature has been added to the INIT_USER routine which disables the RSTNC timeout function based on a jumper input. This helps out ICE™ users since these emulators have difficulty dealing with external resets.

5.7.2 RST USER: SPECIAL HANDLING UPON WARM RESET

When RST_USER is called, you should reset any I/O devices connected to the microcontroller. If dual-port memory is present, rewrite the upper and lower address boundaries to the MPC because the internal MPC registers are all cleared after an iPSB reset. In most cases, the contents of interconnect registers will remain unchanged.

5.7.3 POLL USER: POLLED USER FUNCTIONS

The POLL_USER routine is called from the mainline code. It polls user functions on each loop through the mainline code (approximtely 39 μs). Typical uses are to scan for on-board errors and to set the general error status bit in the General Status Register when errors are found.

5.7.4 USER TIMER: TIMER BASED FUNCTIONS

The USER_TIMER routine is called every 1 ms as part of timer 0 interrupt routine. USER_TIMER is similar to the POLL_USER, except that the granularity of the timer is predictable since the timer 0 interrupt has the highest priority (other than reset). The USER_TIMER routine is used for self-toggling resets and other time based functions. A software prescale counter can be maintained for timing longer intervals.

5.8 GENERATING THE OBJECT MODULE

The final step in code preparation is to assemble all of your new user routines with ASM-51 and then link to the core module using the RL51 linker. Two batch files are supplied to automate the firmware generation process. To use these files, the current directory must be the \ICFW\SRC subdirectory. Here you will find three files with a .MOD file extension. These are the default module, the tables, and your user code. To assemble these modules enter:

```
ASMMOD dft
ASMMOD table
ASMMOD user
```

The list files will be placed in the \ICFW\SRC\LST subdirectory and the object files are placed in the \ICFW\SRC\OBJ subdirectory. Next, the object modules will be linked with the core library to produce a PROM image. Select a name for the output file (e.g., TEST), and invoke the linker by entering:

You will now find a PROMmable file with a .LNK extension in the \ICFW\SRC\OBJ subdirectory. There will also be a file in this subdirectory with a .MAP extension. This file (a link map) will be used in the next step to verify microcontroller resources.

5.8.1 CHECKING MICROCONTROLLER RESOURCE UTILIZATION

The link map file contains information about how much code and register space was requested by the core and user routines and where symbolic variables are located. Examine the link map to check that the limits on RAM and ROM usage have not been exceeded. A gap will be reported whenever additional free space is available. The core firmware will create the following segments:

LNKUSR file name Link Map for test. LNK (Main)

```
TYPE   BASE     LENGTH   RELOCATION  SEGMENT NAME

REG    0000H    0008H                "REG BANK 0"
DATA   0008H    000DH    UNIT        RW_SEG
DATA   0015H    0008H    UNIT        DATA_SEG
       001DH    0003H                *** GAP ***
DATA   0020H    0002H    BIT_ADDR    BIT_ADDR_BYTE_SEG
BIT    0022H    0000H.2  UNIT        BIT_SEG
       0022H.2  0000H.6              *** GAP ***
IDATA  0023H    0008H    UNIT        STACK_SEG

CODE   0000H    004EH    ABSOLUTE
CODE   004EH    0433H    UNIT        CODE_SEG
CODE   0481H    00BFH    UNIT        EDIT_SEG
CODE   0540H    0067H    UNIT        WRITE_SEG
CODE   05A7H    005AH    UNIT        READ_SEG
CODE   0601H    000AH    UNIT        USER_CSEG
       060BH    09F3H                *** GAP ***
CODE   0FFEH    0002H    ABSOLUTE
```

5.8.2 CHECKING CRITICAL TIMING PATHS

Figure 5-4 shows a state diagram of the microcontroller core firmware and includes the instruction cycle counts for each of the major functions. In this step, you will evaluate the impact user code has on interconnect subsystem timing. All timing calculations are based on counting microcontroller machine cycles to compensate for the different crystal frequencies that may be used. To calculate the elapsed time, multiply the cycle count by 1/xtal frequency (i.e., 0.909 μs at 11 MHz). Some microprocessors have specific timing requirements for the pulse width of reset and interrupt signals. These should be checked against the timing values given below:

Reset Timing:

Max RES Latency from UCINT = 41 Cycles

Min RES Inactive after UCINT Inactive = 38 Cycles + RST_USER

NMI Timing:

Max Interrupt Latency = Mainline + IC Handler = 39 + 136 Cycles

Min Interrupt Latency = Mainline = 39 Cycles

The timer 0 interrupt occurs every 1 ms when using the recommended 11 MHz crystal oscillator. However, the time 0 interrupt is masked in both the IC handler and the mainline code. Therefore, the worst case timer 0 latency is calculated:

Max Timer 0 Latency = IC Handler + Mainline = 136 + 49 Cycles

An iPSB timeout will occur if an incoming interconnect request is not completed within 1 ms after **SC0*** becomes active. Consider the case where a local interconnect request is received just a moment before the iPSB request. The microcontroller might be in a timer 0 interrupt routine and would have to pass again through mainline code before servicing the local request. Thus, the worst case iPSB response can be calculated:

Worst Case iPSB Response = Mainline + Timer_0 + Mainline + Local IC Access + Mainline + PSB Access

For the core firmware without user code, the worst case local firmware response is approximately 485 μs at 11 MHz. This leaves a total of 515 μs for the iPSB interconnect cycle to complete (including MPC and write data delays) before a timeout occurs.

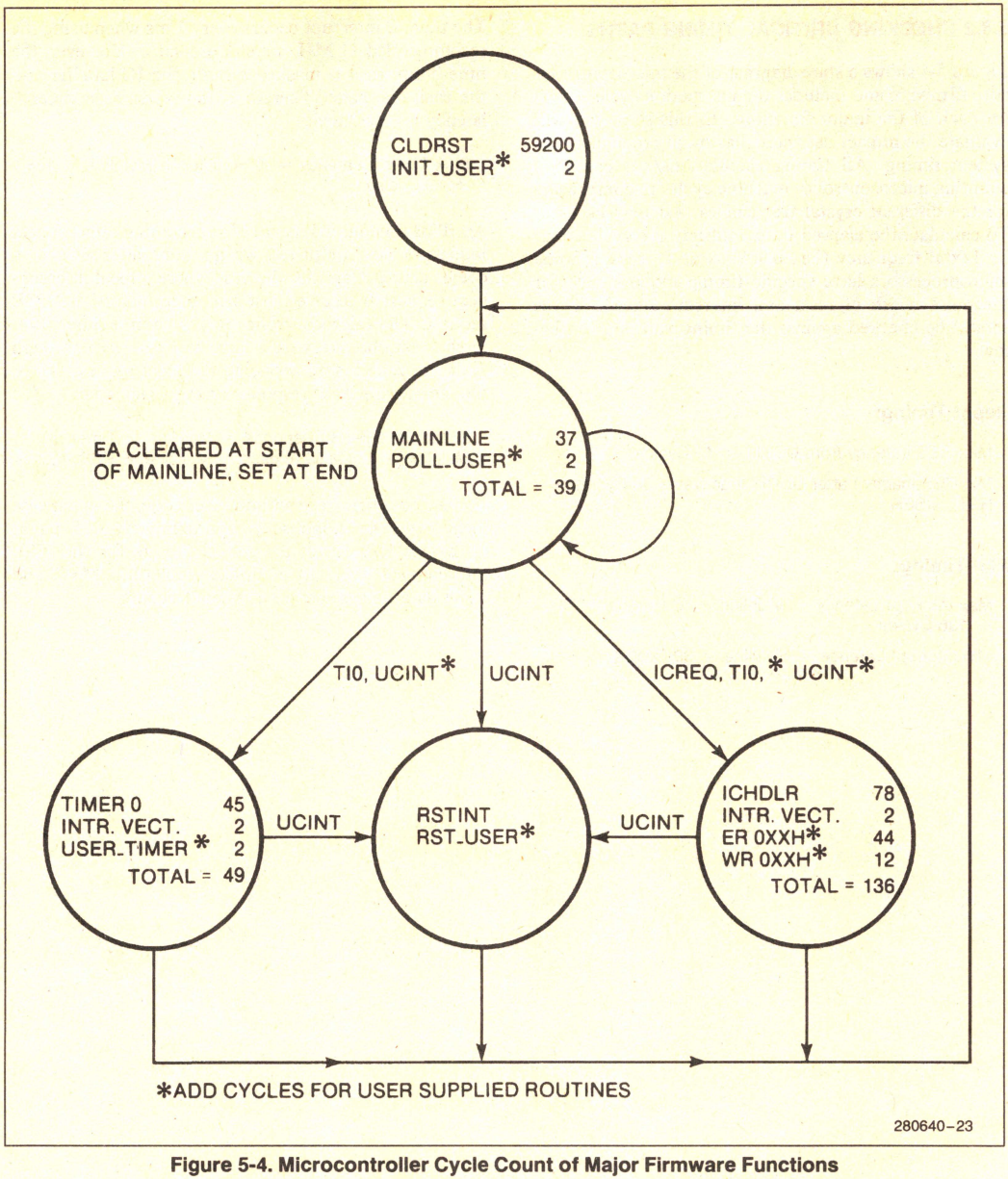


Figure 5-4. Microcontroller Cycle Count of Major Firmware Functions

5.8.3 PROGRAMMING THE 8751

One of the three initialization self-checks that the microcontroller performs following a reset is the PROM check sum test. The firmware computes a 16-bit sum of all code bytes from 0 through (top of the PROM memory-2). The actual check sum is the two's complement of the sum. This value is programmed into the last two bytes of PROM space and given the label CHECKSUM. The check sum label is assigned in the default program file \ICFW\SRC\DFT.MOD and is originally set to address 0FFEH. The address of this label can be modified to accommodate other microcontrollers with larger EPROM size (e.g., an 8752 Microcontroller), as long as the check sum always occupies the last two bytes of code space.

The check sum is initially assembled with a data value of 0000H. After assembling and linking the interconnect firmware, the actual check sum value will be programmed into the PROM in a separate step. The following procedure assumes that an Intel iUP 201 PROM Programmer and iPPS software used. If you are using another brand of PROM programmer, you must guarantee that their check sum algorithm matches the method used here.

Step 1. Invoke iPPS.

Step 2. Initialize default base to hex and the file format to 80.

I 80

Step 3. Set PROM type to 8751.

T 8751

Step 4. Load data buffer with 0FFH.

L B W 0FFH

Step 5. Copy object file to buffer.iPPS will display a check sum.

C<file name>T B

Step 6. Copy check sum displayed by iPPS into address 0FFEH. The low byte of the 16-bit check sum loads to address 0FFEH and the high byte to address 0FFFH.

S 0FFEH
0FFE:(low byte) (high byte)

Step 7. Save a copy of the complete object file on your hard disk

C B T<file name>

Step 8. Install a blank 8751 in the programmer and copy the buffer to PROM.

C B T P

At this point you will have a programmed microcontroller, which is ready to install on the prototype board for functional testing. As an alterntive, you can load the file created in step 7 above into an ICE 51 In-Circuit Emulator and test the firmware directly without programming the 8751. This method gives you access to internal variables and tables in addition to hardware control of the microcontroller.

6.0 FUNCTIONAL TESTING OF THE INTERCONNECT SUBSYSTEMS

6.1 Testing the Interconnect Subsystem

Assuming you were successful in Sections 4 and 5, you now have a functional prototype of your interconnect subsystem ready for testing. Proper interconnect operation depends on complex interrelationships between the 8751 Microcontroller, CPU, Message Passing Coprocessor (MPC), and various TTL circuits. Since interconnect forms the basis of higher level diagnostic services, we must have complete confidence that the interconnect subsystem is functioning properly before implementing the design. Evaluation of the interconnect subsystem is facilitated through the use of an interconnect utility program and a systematic approach to design testing.

6.2 Power-On and Initialization Functions

When a prototype board is inserted into the backplane for the first time, there is a distinct possibility that a design error in the reset initialization sequence could cause a bus error or Reset-Not-Complete (RSTNC) condition. Either of these conditions prevents other boards from coming on-line. Fortunately, these events can usually be detected by watching the front panel LEDs cycle through the power-on sequence.

Under normal reset conditions, the majority of Multibus II boards first illuminate their yellow BIST LED, progress on to illuminate the green LED (run indicator) and leave no other LEDs lit. If the Central Services Module (CSM) has a BUSERR indicator, it will be lit to indicate a bus error due to RSTNC condition. If the bus error persists for more than a few seconds, then a strong likelihood exists that your board has failed to clear the RSTNC* bit. This is confirmed if, after 30 seconds, the CSM BUSERR condition disappears due to the microcontroller clamping reset active and forcing a reset complete indication to the iPSB (unless disabled using the ICE compatibility jumper).

6.3 The Interconnect Configuration Utility

In the \ICFW\ICU subdirectory is an interconnect (IC.P28) written in PL/M source code to run on the iRMX 86/286 Operating Systems. The executable

object file for this utility may be obtained from the iRMX Users Group (iRUG), 5200 N.E. Elam Young Parkway, Hillsboro, OR 97124-9987. If you are not working in one of these programming environments, you will need to adapt this code to your own environment by substituting the appropriate I/O and interconnect system calls.

Before invoking the IC utility on your prototype, you must create the subdirectory /USER/BOARDS and enter into that directory a file for the board. The file name you assign must correspond with the ASCII description contained in the 10 character board ID field of the Interconnect Header Record. Slashes (/) present in the description must be converted to ASCII periods. For example, iSBC386/100 is converted to iSBC386.100. The actual values for each interconnect register are entered into the template plus any user comments you may wish to add for that board. A typical configuration file listing is shown in Figure 6-1. The control characters that appear in the file listing are used by the utility to mark the display area and field boundaries.

- The @ symbol is the screen delimiter. As the user pages through the file and encounters this symbol, the utility will prompt the user for input.
- The ; symbol is the register field delimiter. The utility scans for an opening semicolon; then fetches a register value for each character found until encountering the closing semicolon. Legal character type descriptors are:

  b (binary)

  d (decimal)

  h (hexadecimal)

  a (ASCII)

The IC utility enables you to identify, examine, and modify the configuration of a system board from the comfort of your console (no jumpers to move!). The utility locates the board ID character fields in the Interconnect Header Record of the board in the cardslot you select to configure. It then searches the subdirectory /USER/BOARDS for the file having a file name that matches the board ID. Speciic interconnect registers may be modified (provided access rights and value checking are valid), however, an attempt to enter illegal values into an interconnect register are ignored and the existing register content is unaffected. Entries are displayed as they are made. Once invoked, the IC utility prompts the user for input and you will find it very easy to use. A sample screen output from this utility is shown in Figure 6-2.

6.4 A Test Methodology for Interconnect Functions

Given a functional prototype board and an interconnect utility such as the one described above, we now use these tools to verify correct operation of the interconnect subsystem. One might assume that this process would be quite simple and straightforward—and usually, it is. But we are now looking for more subtle conditions such as timing related problems and secondary affects on interconnect register content. While every board is different, a test methodology that looks for all possible contingencies is an important step in design verification. The following checklist is an example of what to look for during evaluation testing of interconnect using a prototype board:

Look for initialization conditions:

Do the correct default register values appear after cold-start?

Do the front panel LEDs work properly?

Is RSTNC handled correctly? (also test RSTNC recovery by disabling the CPU).

Look for dual-port operation on both local and system buses:

Can the registers be accessed from off-board as well as on-board?

Do the function records all have correct record type values and byte counts?

Are the read/write privileges working for both on-board and off-board references?

Is the protection record function working properly?

Look for proper register content and function:

Do all the bit level functions in control registers work as expected?

Are all the status registers reporting correctly?

Look for secondary effects:

Changes to memory addressing registers are only allowed when memory is disabled.

Look for iPSB timeout conditions:

Saturate the local bus with interconnect I/O commands in a tight loop; then attempt an interconnect operation from off-board.

Once you have accomplished this type of testing methodology for each register on the prototype board, you can have reasonable confidence that all is well with the interconnect subsystem. Now you can proceed forward with the development of device drivers and extended diagnostics. If any changes are made to the microcon-

troller after its initial release, be sure to update the revision control field in the header record so users can identify which version of the firmware they hold.

6.5 Summary and Conclusion

Multibus II is a system architecture composed of standarized hardware and software modules. Having successfully implemented the interconnect subsystem on your board, you are ensured of the functional compatibility and interoperability of that board with other industry-standard Multibus II products that use interconnect.

If you found this guide useful, you may also be interested in other design guides available from Intel. A list of documents available at the time of this guide's publication are listed in Appendix A. As the Multibus II system architecture matures and proliferates, we look forward to supplying additional design guides that encourage the production of compatible hardware and software products.

```
Function                    Value  RW  Comments
Vendor ID Number             ;h h ; ro
Board ID Number         ;aaaaaaaaaa; ro
PBA Number, Rev          ;d d d d ; ro
Hardware Test Rev #           ;d ; ro
Class ID                      ;b ; ro
Reserved                 ;b b b b ; ro
Program Table Index           ;b ; rw
NMI Enable Register           ;b ; rw
General Status                ;b ; ro
General Control               ;b ; rw
BIST Support Level            ;b ; rw
BIST Data In                  ;b ; rw
BIST Data Out                 ;b ; rw
BIST Slave Status             ;b ; rw
BIST Master Status            ;b ; rw
BIST Test ID                  ;b ; rw

Protection Record           Value  RW  Comments
Protection Record Type        ;b ; ro
Record Length                 ;b ; ro
Protection Level Reg          ;b ; rw
Reserved                      ;b ; ro

Memory Record Type          Value  RW  Comments
Memory Record Type            ;b ; ro
Record Length                 ;b ; ro
Memory Size (bits 7..0)       ;b ; ro
Memory Size (bits 15..8)      ;b ; ro
Memory Control                ;b ; rw
Memory Status                 ;b ; ro
MM2 ID                        ;b ; ro

iPSB Control Record         Value  RW  Comments
iPSB Control Record Type      ;b ; ro
Record Length                 ;b ; ro
iPSB Slot ID Number           ;b ; ro
iPSB Arbitration ID Number    ;b ; rw
iPSB Error Register           ;b ; rw
iPSB Control/Status Register  ;b ; rw
iPSB Diagnostic Register      ;b ; rw
Reserved                      ;b ; ro
```

280640-24

Figure 6-1. Configuration File Format for the IC Utility (Sheet 1 of 3)

```
Reg   iPSB Memory Record Type     Value  RW  Comments
33    iPSB Memory Record Type       ;b ; ro
34    Record Length                 ;b ; ro
35    iPSB Start Address (23..16)   ;b ; rw
36    iPSB Start Address (31..24)   ;b ; rw
37    iPSB End Address (23..16)     ;b ; rw
38    iPSB End Address (31..24)     ;b ; rw
39    iPSB Memory Control           ;b ; rw
@
Reg   Local Memory Record Type    Value  RW  Comments
3A    Local Memory Record Type      ;b ; ro
3B    Record Length                 ;b ; ro
3C    Local Start Address (23..16)  ;b ; ro
3D    Local Start Address (31..24)  ;b ; ro
3E    Local End Address (23..16)    ;b ; rw
3F    Local End Address (31..24)    ;b ; ro
40    Local Control Register        ;b ; rw
@
Reg   Memory Parity Record Type   Value  RW  Comments
41    Memory Parity Record Type     ;b ; ro
42    Record Length                 ;b ; ro
43    Parity Control Register       ;b ; rw
44    Parity Status Register        ;b ; ro
45    Bank Status Register          ;b ; ro
46    Error Offset             ;b b b b ; ro
4A    Reserved                      ;b ; ro
@
Reg   Cache Memory Record         Value  RW  Comments
4B    Cache Memory Record Type      ;b ; ro
4C    Record Length                 ;b ; ro
4D    Cache Size (bits 7..0)        ;b ; ro
4E    Cache Size (bits 15..8)       ;b ; ro
4F    Cache Entry Size              ;b ; ro
50    Cache Control                 ;b ; rw
51    Reserved                      ;b ; ro
@
```

280640-25

Figure 6-1. Configuration File Format for the IC Utility (Sheet 2 of 3)

5

```
Reg     Firmware Comm Record Type   Value  RW   Comments
52      Firmware Comm Record Type     ;b ; ro
53      Record Length                 ;b ; ro
54      Communications Byte 1         ;b ; rw
55      Communications Byte 2         ;b ; rw
56      Communications Byte 3         ;b ; rw
57      Communications Byte 4         ;b ; rw
58      Communications Byte 5         ;b ; rw
59      Communications Byte 6         ;b ; rw
5A      Communications Byte 7         ;b ; rw
5B      Communications Byte 8         ;b ; rw
5C      Communications Byte 9         ;b ; rw
5D      Communications Byte 10        ;b ; rw
5E      Communications Byte 11        ;b ; rw
5F      Communications Byte 12        ;b ; rw
60      Communications Byte 13        ;b ; rw
61      Communications Byte 14        ;b ; rw
62      Communications Byte 15        ;b ; rw
63      Communications Byte 16        ;b ; rw
@
Reg     Host ID Record Type         Value  RW   Comments
64      Host ID Record Type           ;b ; ro
65      Record Length                 ;b ; ro
66      Host ID (bits 7..0)           ;b ; rw
67      Host ID (bits 15..8)          ;b ; rw
68      Message Address               ;b ; rw
69      Reserved                      ;b ; ro
@
Reg     Serial Comm Record Type     Value  RW   Comments
6A      Serial Comm Record Type       ;b ; ro
6B      Record Length                 ;b ; ro
6C      Serial Data In                ;b ; ro
6D      Serial Data Out               ;b ; rw
6E      Serial Port Status            ;b ; ro
6F      Serial Interrupt Enable       ;b ; rw
70      Serial Port Options           ;b ; rw
71      Reserved                      ;b ; ro
@
Reg     386/100 Specific Record     Value  RW   Comments
72      386/100 Spec Record Type      ;b ; ro
73      Record Length                 ;b ; ro
74      On-board Control Register     ;b ; rw
@
Reg     End Of Template Record      Value  RW   Comments
75      EOT Record Type               ;b ; ro
@
```

280640-26

Figure 6-1. Configuration File Format for the IC Utility (Sheet 3 of 3)

```
            INTERCONNECT CONFIGURATION UTILITY

Commencing Board Search Routine:  .........
Board Search Complete.  The following boards were found:

Slot#  Vendor ID          Board ID    PBA #         Class
00     Intel Corporation  CSM/001     147304-0003   Central Services Module
01     Intel Corporation  186/410     000000-0002   16-bit Communications Board
02*    Intel Corporation  386/100     000000-0002   32-bit Processor Board
03
04     Intel Corporation  MEM/310     000000-0001   Memory Board
05     Intel Corporation  286/100A    000000-0001   16-bit Processor Board
06
07     Intel Corporation  186/224A    000000-0001   16-bit Peripheral Controller
08

Options: <Configure,Slot#>, <Memory> or <Quit> Type First Letter):c2

Configuration File Attached - File: /user/boards/386.100

Reg    Function                    Value  RW   Comments
00     Vendor ID Number             0100; ro
02     Board ID Number            386/100; ro
0C     PBA Number, Rev           00000000; ro
10     Hardware Test Rev #            02; ro
11     Class ID                       F1; ro
12     Reserved                  00000080; ro
16     Program Table Index            00; rw
17     NMI Enable Register            04; rw
18     General Status                 80; ro
19     General Control                00; rw
1A     BIST Support Level             00; rw
1B     BIST Data In                   00; rw
1C     BIST Data Out                  00; rw
1D     BIST Slave Status              10; rw
1E     BIST Master Status             30; rw
1F     BIST Test ID                   10; rw

Would you like to make any changes?

Enter <reg_num,new_value_in_hex>,<Quit>,<CR> or <Up>:
```

280640-27

Figure 6-2. Interconnect Utility Screens (Sheet 1 of 2)

```
Reg    Protection Record          Value  RW       Comments
20     Protection Record Type       08; ro
21     Record Length                02; ro
22     Protection Level Reg         00; rw
23     Reserved                     00; ro

Reg    Memory Record Type         Value  RW       Comments
24     Memory Record Type           01; ro
25     Record Length                05; ro
26     Memory Size (bits 7..0)      3F; ro
27     Memory Size (bits 15..8)     00; ro
28     Memory Control               01; rw
29     Memory Status                A1; ro
2A     MM2 ID                       22; ro

Would you like to make any changes?

Enter <reg_num,new_value_in_hex>,<Quit>,<CR> or <Up>:

Reg    iPSB Control Record        Value  RW       Comments
2B     iPSB Control Record Type     06; ro
2C     Record Length                06; ro
2D     iPSB Slot ID Number          10; ro
2E     iPSB Arbitration ID Number   E8; rw
2F     iPSB Error Register          88; rw
30     iPSB Control/Status Register 18; rw
31     iPSB Diagnostic Register     00; rw
32     Reserved                     00; ro

Would you like to make any changes?

Enter <reg_num,new_value_in_hex>,<Quit>,<CR> or <Up>:q
```

280640-28

Figure 6-2. Interconnect Utility Screens (Sheet 2 of 2) (Continued)

APPENDIX A
BIBLIOGRAPHY OF RELATED READING

Interconnect Architecture Specification—Intel Order Number 149299-002.

MPC User's Guide—Intel Order Number 176525-001.

IEEE 1296 Specification (High Performance Synchronous 32-Bit Bus Standard)

iSBC 386/116 Hardware Reference Manual—Intel Order Number 451833-001.

BUSCON Paper: An Architecture for Initializing Multibus II Multiprocessor Systems. Stephen Rogers, October 1987.

Computer Technology Review Quarterly: Interconnect Simplifies Configuration. Roger Finger, September 1987.

Intel Application Note AP-70, Using the Intel MCS® 51 Boolean Processing Capabilities. John Wharton, 1980.

Embedded Controller Handbook 8751 Data Sheet—Intel Order Number 210918-005.

iUP 200/201A Programmer User's Guide—Intel Order Number 166608-001.

MCS Macro Assembler User's Guide for DOS Systems—Intel Order Number 122752-001.

MCS 51 Utilities User's Guide for DOS Systems—Intel Order Number 122747-001.

Intel Application Note AP-422, Designing a Central Services Module for Multibus II. Jory Radke, 1987.

APPENDIX B
FLOWCHARTS FOR IC CORE FIRMWARE

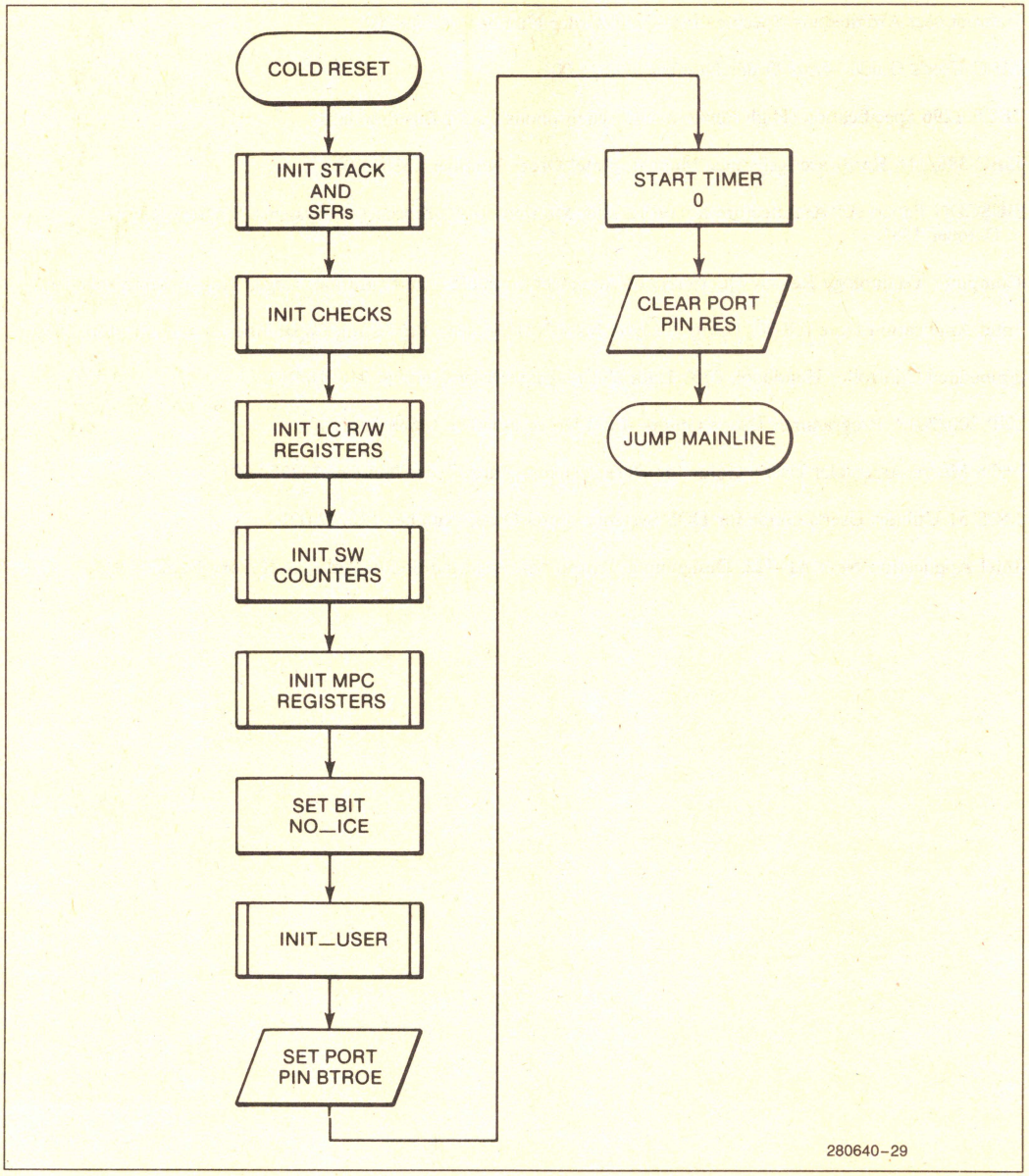


Figure B-1. Cold Reset Routine

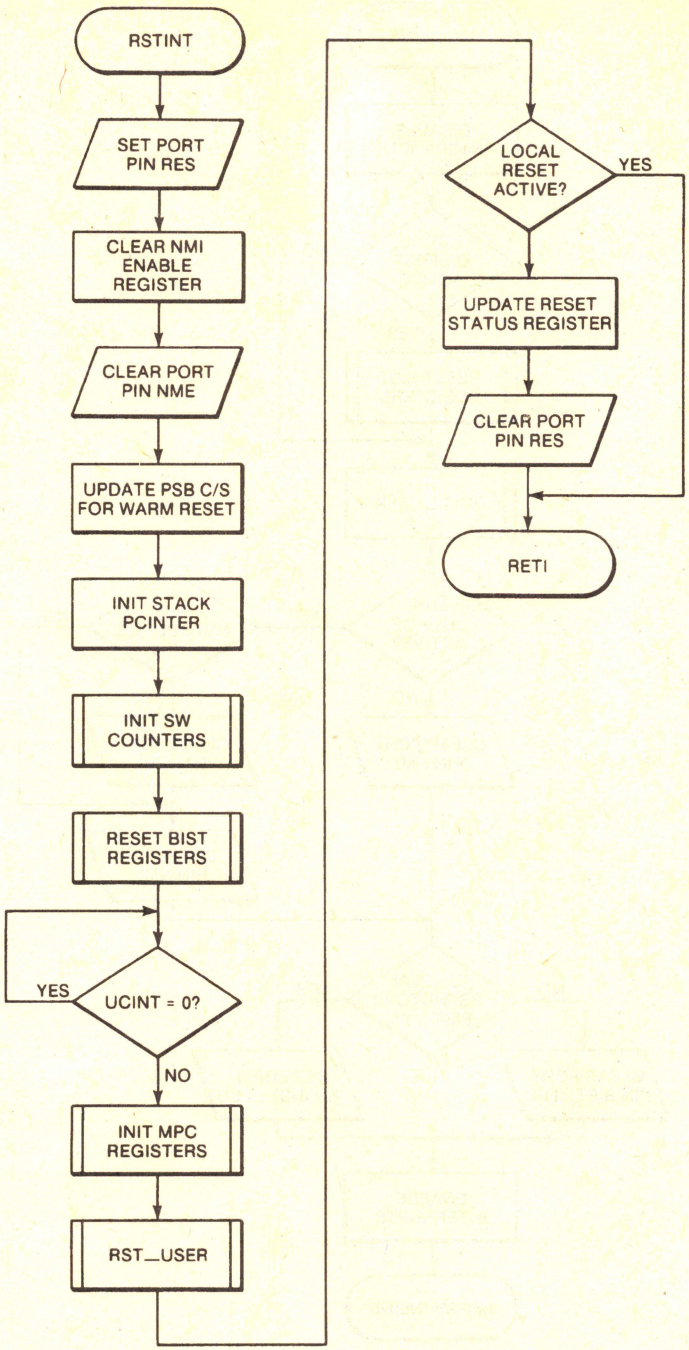


Figure B-2. Reset Interrupt Routine

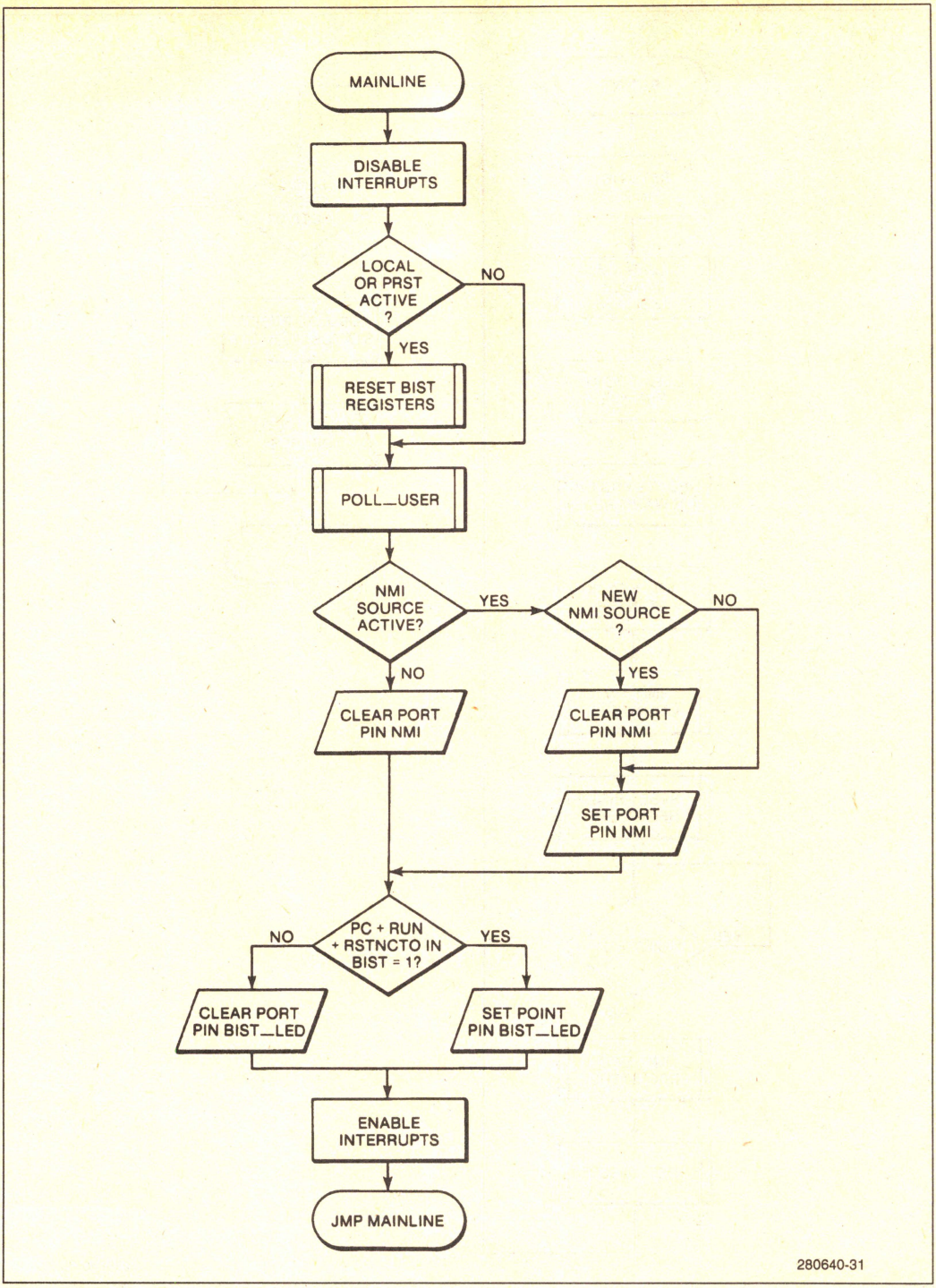


Figure B-3. Mainline Program Flow

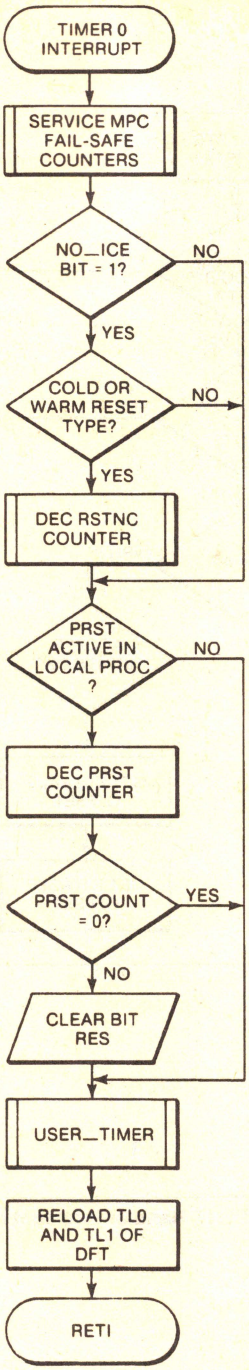


Figure B-4. Timer 0 Interrupt Routine

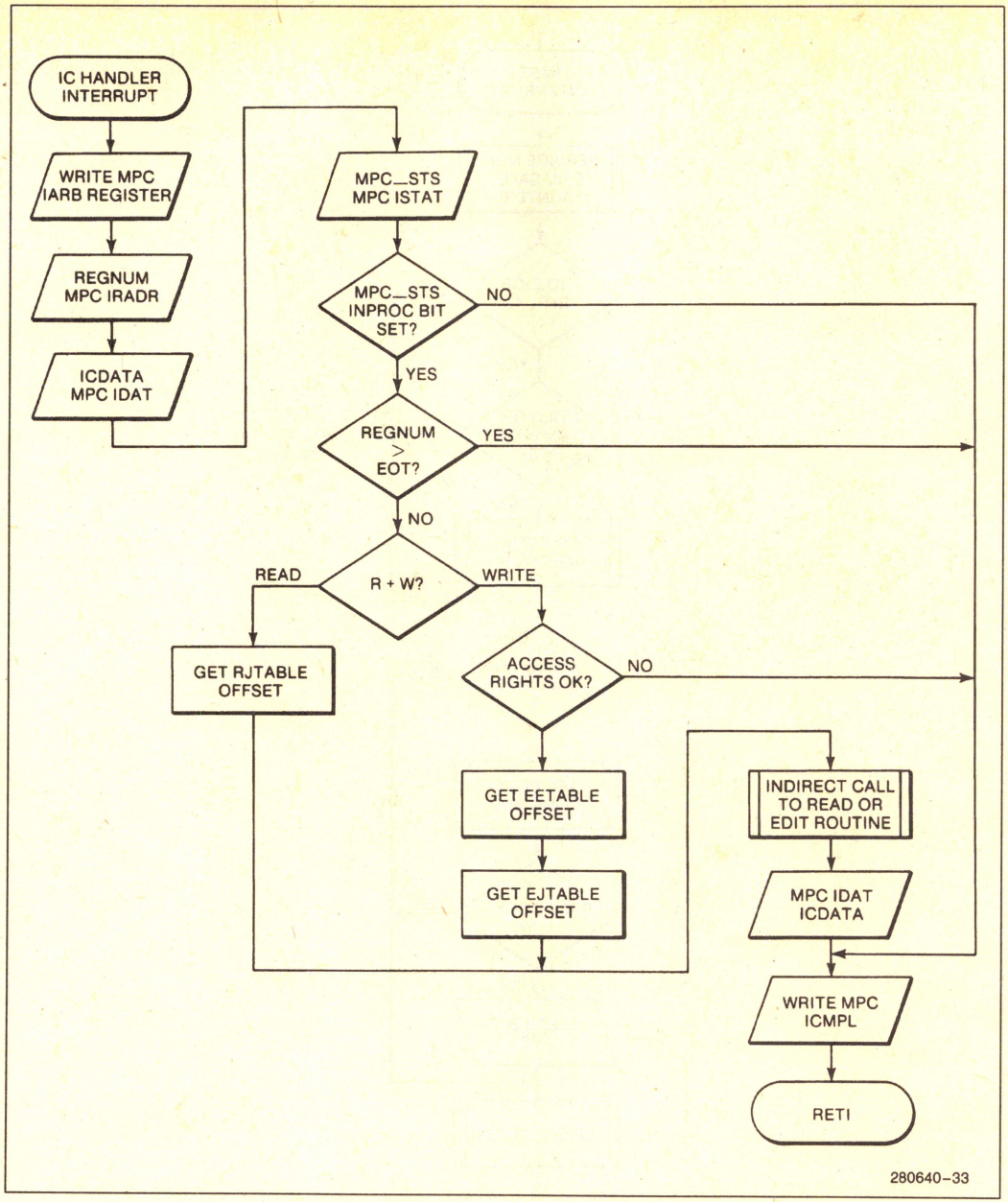


Figure B-5. IC Handler Interrupt Routine

APPENDIX C
READ, WRITE, AND EDIT ROUTINES

```
;*******************************************************
;
;  NAME:          IC_TO_MPC
;
;  PURPOSE:       CORRELATE IC REGISTER NUMBER TO MPC
;                 REGISTER NUMBER
;
;  DESCRIPTION:   CORRELATES IC PSB C/S REGISTERS WITH
;                 THE PROPER MPC REGISTER NUMBER,
;                 STARTING WITH THE PSB SLOT ID
;                 REGISTER.
;
;  UPDATE HISTORY:
;
;*******************************************************
;
IC_TO_MPC:  DB      SID         ;PSB SLOT ID
            DB      AID         ;PSB ARB ID
            DB      RERR        ;PSB ERROR
            DB      00H         ;DUMMY
            DB      DIAG        ;PSB DIAGNOSTIC
$EJECT
```

280640-34

Figure C-1. IC__TO__MPC Corrrelation

```
;*******************************************************
;
;  NAME:          RD00H
;
;  PURPOSE:       READ DEFAULT VALUE FROM DFTABLE
;
;  CALLED BY:     JUMPED TO FROM SERVICE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   USES REGNUM TO INDEX INTO DFTABLE.
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        REGNUM
;
;  DESTROYS:      ACC, DPTR
;
;  RETURNS:       ICDATA
;
;  UPDATE HISTORY:
;
;*******************************************************

RD00H:  %LOOKUP(#DFTABLE,REGNUM) ;GET DEFAULT FM TABLE
        MOV     ICDATA,A         ;RET DEFAULT IN ICDATA
        RET                      ;RETURN TO IC HANDLER
$EJECT
```

280640-35

Figure C-2. RD00H Read DFTABLE

```
;*******************************************************
;
;  NAME:          RD01H
;
;  PURPOSE:       READ RWTABLE VALUE
;
;  CALLED BY:     JUMPED TO FROM SERVICE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   USES REGNUM TO INDEX INTO RATABLE.
;                 USES RWTABLE INDEX PORTION OF RA BYTE
;                 TO INDEX INTO RWTABLE TO GET REGISTER
;                 VALUE.
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        REGNUM
;
;  DESTROYS:      ACC, ADDR, DPTR
;
;  RETURNS:       ICDATA
;
;  UPDATE HISTORY:
;
;*******************************************************

RD01H:  %LOOKUP(#RATABLE,REGNUM)  ;GET RAM/ACCESS BYTE
        CLR     ACC.GAR ;MASK OFF GLOBAL ACCESS RIGHTS
        ADD     A,#RWTABLE ;ADD TBL ADDR TO RAM OFFSET
        MOV     ADDR,A                  ;SAVE RAM ADDR
        MOV     A,@ADDR                 ;READ RAM VALUE
        MOV     ICDATA,A        ;RETURN VALUE IN ICDATA
        RET                       ;RETURN TO IC HANDLER

$EJECT
```

280640-36

Figure C-3. RD01H Read RWTABLE

```
;*******************************************************
;
;  NAME:          RD02H
;
;  PURPOSE:       READ MPC REGISTER FOR PSB
;                 CONTROL/STATUS REGISTER
;
;  CALLED BY:     JUMPED TO FROM SERVICE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   USES LOOKUP TABLE IC_TO_MPC TO XREF
;                 IC ADDRESS TO MPC REGISTER NUMBER,
;                 READS MPC REG AND RETURNS DATA IN
;                 ICDATA.
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        NONE
;
;  DESTROYS:      ACC, DPTR, MPC_RNUM
;
;  RETURNS:       ICDATA
;
;  UPDATE HISTORY:
;
;*******************************************************

RD02H:  MOV     A,REGNUM    ;GET REG NO. BEING ACCESSED
        SUBB    A,#PSB_CTL_OFF+5     ;IC_TO_MPC STARTS
                                     ;WITH SLOT ID REG
        %LOOKUP(#IC_TO_MPC,A)  ;GET CORRES MPC REG NO.
        %RD_MPC(ICDATA,A)              ;READ MPC REG
        RET                      ;RETURN TO IC HANDLER

$EJECT
```

280640-37

Figure C-4. RD02H Read MPC Register

```
;*******************************************************
;
;  NAME:          RD03H
;
;  PURPOSE:       READ ROUTINE FOR GENERAL STATUS
;                 REGISTER
;
;  CALLED BY:     JUMPED TO FROM SERVICE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   USES LOOKUP TABLE IC_TO_MPC TO XREF
;                 IC ADDRESS TO MPC REGISTER NUMBER,
;                 READS MPC REG AND RETURNS DATA IN
;                 ICDATA.
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        NONE
;
;  DESTROYS:      ACC, DPTR, MPC_RNUM
;
;  RETURNS:       ICDATA
;
;  UPDATE HISTORY:
;
;*******************************************************

RD03H:  MOV     ICDATA,GEN_STS              ;READ RAM
        %MOVBIT(ICDATA.PFI,DCLO)  ;GET PWR FAIL STATUS
        RET                       ;RETURN TO IC HANDLER

$EJECT
```

280640-38

Figure C-5. RD03H Read General Status Register

```
;*******************************************************

;  NAME:          RD04H
;
;  PURPOSE:       READ ROUTINE FOR PROTECTION LEVEL
;                 REGISTER
;
;  CALLED BY:     JUMPED TO FROM SERVICE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   READS THE BIT UACRS AND PUTS IT IN
;                 ICDATA
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        NONE
;
;  DESTROYS:      CARRY FLAG
;
;  RETURNS:       ICDATA
;
;  UPDATE HISTORY:
;
;*******************************************************

RD04H:  MOV     ICDATA,#CLEAR               ;ICDATA = 0
        %MOVBIT(ICDATA.ACRS,UACRS)            ;GET BIT
        RET                         ;RETURN TO IC_HANDLER

$EJECT
```

280640-39

Figure C-6. RD04H Read Protection Level Register

```
;*******************************************************
;
;  NAME:          RD05H
;
;  PURPOSE:       READ ROUTINE FOR PSB CONTROL/STATUS
;                 REGISTER
;
;  CALLED BY:     JUMPED TO FROM SERVICE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   COMBINES VALUES OF MPC GENERAL
;                 PARAMETER REGISTER WITH VALUE
;                 FROM RWTABLE FOR PSB_CS TO GET
;                 COMPLETE REGISTER VALUE.
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        NONE
;
;  DESTROYS:      ACC, MPC_RNUM, CARRY FLAG
;
;  RETURNS:       ICDATA
;
;  UPDATE HISTORY:
;
;*******************************************************

RD05H:  %RD_MPC(A,#GEN_PAR)   ;READ MPC GENL PARAM REG
        MOV     ICDATA,PSB_CS                ;READ RAM
        %MOVBIT(ICDATA.HPRO,ACC.PRY)  ;MOV MPC BITS TO
                                              ;ICDATA
        %MOVBIT(ICDATA.RSTNC,ACC.RSTNCIN)
        RET                       ;RETURN TO IC_HANDLER

$EJECT
```

280640-40

Figure C-7. RD05H Read Control/Status Register

```
;*******************************************************
;
;  NAME:          RD06H
;
;  PURPOSE:       READ ROUTINE FOR LOCAL PROC CONTROL
;                 REGISTER
;
;  CALLED BY:     JUMPED TO FROM SERVICE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   IF PROC RESET AND RESET STATUS =
;                 PRST, THEN SETS PRST = TRUE, ELSE
;                 PRST = FALSE.
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        NONE
;
;  DESTROYS:      CARRY FLAG
;
;  RETURNS:       ICDATA
;
;  UPDATE HISTORY:  5-14-87 JR UPDATED FOR IDX
;
;*******************************************************

RD06H:  MOV   ICDATA,#CLEAR                ;ICDATA = 0
        JNB   RES,END_R6  ;IF NOT PROC RESET THEN DONE
        MOV   A,RST_STS       ;ELSE GET LAST RESET TYPE
        CJNE  A,#PRST_TYPE,END_R6     ;IF NOT PRST THEN
                                                ;DONE
        SETB  PRST                ;ELSE SET PRST = TRUE
END_R6: RET                       ;RETURN TO IC HANDLER

$EJECT
```

280640-41

Figure C-8. RD06H Read Local Processor Control Register

```
;*******************************************************
;
;  NAME:          RD07H
;
;  PURPOSE:       READ ROUTINE FOR LOCAL STATUS
;                 REGISTER
;
;  CALLED BY:     JUMPED TO FROM SERVICE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   IF RESET STATUS = PRST TYPE, THEN
;                 PRST STATUS = TRUE, ELSE PRST
;                 STATUS = FALSE.
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        NONE
;
;  DESTROYS:      CARRY FLAG
;
;  RETURNS:       ICDATA
;
;  UPDATE HISTORY:  5-14-87 JR ADDED FOR IDX
;
;*******************************************************

RD07H:  MOV     ICDATA,#CLEAR              ;ICDATA = 0.
        MOV     A,RST_STS          ;GET LAST RESET TYPE
        CJNE    A,#PRST_TYPE,END_R7  ;IF NOT PRST THEN
                                                ;DONE
        SETB    ICDATA.PRST_STS  ;ELSE PRST_STS = TRUE
END_R7: RET                        ;RETURN TO IC HANDLER

$EJECT
```

280640-42

Figure C-9. RD07H Read Local Status Register

```
;*******************************************************
;
;  NAME:          WR00H
;
;  PURPOSE:       WRITE TO A RAM BASED IC REGISTER
;
;  CALLED BY:     JUMPED TO FROM EDIT ROUTINES
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   USES REGNUM TO INDEX INTO RATABLE.
;                 USES RWTABLE INDEX PORTION OF RA BYTE
;                 TO INDEX INTO RWTABLE, WHERE ICDATA
;                 IS THEN WRITTEN TO.
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        REGNUM
;
;  DESTROYS:      ACC, DPTR, ADDR
;
;  RETURNS:       NOTHING, RETURNS TO ICHANDLER
;
;  UPDATE HISTORY:
;
;*******************************************************

WR00H:  %LOOKUP(#RATABLE,REGNUM)        ;GET RAM/ACCESS
                                          ;BYTE FOR REG
        CLR     ACC.GAR           ;EXTRACT RWTABLE INDEX
        ADD     A,#RWTABLE  ;ADD TABLE ADDR TO GET RAM
                                                ;ADDR
        MOV     ADDR,A  ;PUT ADDR IN INDIRECT ADDR REG
        MOV     @ADDR,ICDATA            ;WR RWTABLE REG
        RET                         ;RETURN TO IC HANDLER

$EJECT
```

280640-43

Figure C-10. WR00H Write RAM Based IC Register

```
;*******************************************************
;
;  NAME:          WR01H
;
;  PURPOSE:       WRITE ROUTINE FOR MPC BASED IC
;                 REGISTERS
;
;  CALLED BY:     JUMPED TO FROM EDIT ROUTINES
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   USES LOOKUP TABLE IC_TO_MPC TO XREF
;                 IC ADDRESS TO MPC REGISTER NUMBER,
;                 WRITES MPC REG WITH DATA IN ICDATA.
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        ICDATA
;
;  DESTROYS:      DPTR, MPC_RNUM
;
;  RETURNS:       NOTHING, RETURNS TO ICHDLR
;
;  UPDATE HISTORY:
;
;*******************************************************

WR01H:  MOV     A,REGNUM        ;GET REQUESTED REG NO.
        SUBB    A,#PSB_CTL_OFF+5    ;IC_TO_MPC STARTS
                                ;WITH PSB SLOT ID REG
        %LOOKUP(#IC_TO_MPC,A)  ;GET CORRES MPC REG NO.
        %WRT_MPC(A,ICDATA)              ;WR MPC REG
        RET                     ;RETURN TO IC HANDLER

$EJECT
```

280640-44

Figure C-11. WR01H Write MPC Based IC Register

5

```
;*******************************************************
;
;  NAME:          WR02H
;
;  PURPOSE:       WRITE ROUTINE FOR GENERAL CONTROL
;                 REGISTER
;
;  CALLED BY:     JUMPED TO FROM EDIT ROUTINE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   PROC RESET = LOCAL RESET.  IF LOCAL
;                 RESET, THEN UPDATE RESET STATUS REG.
;                 UPDATES GEN_CTRL IN RWTABLE BY
;                 JUMPING TO WR00H.
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        ICDATA
;
;  DESTROYS:      NOTHING
;
;  RETURNS:       NOTHING, JUMPS TO WR00H
;
;  UPDATE HISTORY:  5-14-87 JR UPDATED FOR IDX
;
;*******************************************************

WR02H:  %MOVBIT(RES,ICDATA.LRST)  ;UPDATE RST PORT PIN
        JNC     END_W2    ;IF NO LOCAL RESET THEN DONE
        MOV     RST_STS,#LRST_TYPE  ;ELSE UPDATE RESET
                                                ;STATUS
END_W2: SJMP    WR00H

$EJECT
```

280640-45

Figure C-12. WR02H Write General Control Register

```
;*******************************************************
;
;  NAME:          WR03H
;
;  PURPOSE:       WRITE ROUTINE FOR PROTECTION LEVEL
;                 REGISTER
;
;  CALLED BY:     JUMPED TO FROM EDIT ROUTINE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   SETS DYNAMIC ACCESS RIGHTS AS
;                 REQUESTED
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        ICDATA
;
;  DESTROYS:      CARRY FLAG
;
;  RETURNS:       NOTHING, RETURNS TO ICHDLR
;
;  UPDATE HISTORY:
;
;*******************************************************

WR03H:  %MOVBIT(UACRS,ICDATA.ACRS);UACRS = ICDATA.ACRS
        RET                        ;RETURN TO IC HANDLER

$EJECT
```

280640-46

Figure C-13. WR03H Write Protection Level Register

```
;*******************************************************
;
;  NAME:          WR04H
;
;  PURPOSE:       WRITE ROUTINE FOR PSB CONTROL/STATUS
;                 REGISTER
;
;  CALLED BY:     JUMPED TO FROM EDIT ROUTINE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   SINCE RST TYPE BITS MUST = 0 WHEN
;                 WRITING THIS REGISTER, RSTNCOUT IS
;                 CLEARED IN THE MPC GENERAL PARAM REG.
;                 ALSO,  THE PRY BIT IN THE GEN PARAM
;                 REG IS MODIFIED ACCORDING TO ICDATA
;                 BIT HPRO.  SINCE THE GEN PARAM REG
;                 SUPPORTS OTHER FUNCTIONS, A
;                 READ/MODIFY/WRITE OPERATION IS USED.
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        ICDATA
;
;  DESTROYS:      ACC, CARRY FLAG, MPC_RNUM
;
;  RETURNS:       NOTHING
;
;  UPDATE HISTORY:
;
;*******************************************************

WR04H:  MOV    A,PSB_CS ;GET VALUE OF PSB CONT/STAT REG
        %MOVBIT(ICDATA.RSTH0,ACC.RSTH0)    ;SAVE RESET
                                          ;TYPE HISTORY BITS
        %MOVBIT(ICDATA.RSTH1,ACC.RSTH1)
        MOV    PSB_CS,ICDATA              ;UPDATE RWTABLE
        %RD_MPC(A,#GEN_PAR)              ;READ MPC GEN_PAR
        CLR    ACC.RSTNCOUT      ;QUIT DRVG RSTNC TO BUS
        %MOVBIT(ACC.PRY,ICDATA.HPRO)   ;UPDATE PRY BIT
        %WRT_MPC(#GEN_PAR,A)           ;WR MODIFIED DATA
                                        ;BACK TO GEN_PAR
        RET                          ;RETURN TO IC HANDLER

$EJECT
```

280640-47

Figure C-14. WR04H Write PSB Control/Status Register

```
;*******************************************************
;
;  NAME:          WR05H
;
;  PURPOSE:       WRITE ROUTINE FOR PSB DIAGNOSTIC
;                 REGISTER
;
;  CALLED BY:     JUMPED TO FROM EDIT ROUTINE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   UPDATES PSB_DIAG REG IN RWTABLE.  IF
;                 LOOPBACK MODE SELECTED, THEN TURNS
;                 OFF BUS TRANSCEIVERS AND WRITES MPC
;                 DIAG REG, ELSE WRITES MPC DIAG REG
;                 AND ENABLES TRANSCEIVERS.
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  DESTROYS:      ACC, MPC_RNUM
;
;  INPUTS:        ICDATA
;
;  RETURNS:       NOTHING, RETURNS TO ICHDLR
;
;  UPDATE HISTORY:
;
;*******************************************************

WR05H:  MOV    PSB_DIAG,ICDATA          ;UPDATE RWTABLE
        JB     ICDATA.LBACK,BUSOFF        ;IF LOOPBACK
                                  ;SELECTED, THEN JUMP
        %WRT_MPC(#DIAG,ICDATA)      ;ELSE WR MPC REG
        SETB   BTROE                ;ENABLE BUS XCVRS
        JMP    END_W5                    ;DO NO MORE
BUSOFF: CLR    BTROE                  ;DISABLE XCVRS
        %WRT_MPC(#DIAG,ICDATA)           ;WR MPC REG
END_W5: RET                       ;RETURN TO IC HANDLER

$EJECT
```

280640-48

Figure C-15. WR05H Write PSB Diagnostic Register

5

```
;*******************************************************
;
;  NAME:          WR06H
;
;  PURPOSE:       WRITE ROUTINE FOR LOCAL PROC CONTROL
;                 REGISTER
;
;  CALLED BY:     JUMPED TO FROM EDIT ROUTINE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   IF ICDATA BIT PRST SET, ASSERT RESET
;                 TO PROC AND UPDATE RESET STATUS REG.
;
;  INPUTS:        ICDATA
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  DESTROYS:      CARRY FLAG
;
;  RETURNS:       NOTHING
;
;  UPDATE HISTORY:  5-14-87 JR ADDED FOR IDX
;
;*******************************************************

WR06H:  JNB    ICDATA.PRST,END_W6;IF NO RESET THEN DONE
        SETB   RES             ;ELSE ASSERT RESET TO PROC
        MOV    RST_STS,#PRST_TYPE          ;UPDATE RESET
                                             ;STATUS REG
        MOV    PRST_CNTR,#PRST_CNT   ;LOAD PRST COUNTER
END_W6: RET                         ;RETURN TO IC HANDLER

$EJECT
```

280640-49

Figure C-16. WR06H Write Local Processor Control Register

```
;*******************************************************
;
;  MACRO NAME:    GET_EEC (GET EDIT ENTRY CONSTANT)
;
;  SYNTAX:        GET_EEC
;
;  PURPOSE:       INDEX INTO EETABLE TO GET NEXT EDIT
;                 ENTRY CONSTANT
;
;  DESCRIPTION:   EET_OFF IS INCREMENTED AND THE CODE
;                 BYTE AT EET_OFF FROM DPTR (ASSUMED TO
;                 BE POINTING TO EETABLE) IS MOVED TO
;                 THE ACCUMULATOR.
;
;  INPUTS:        NONE, HOWEVER DPTR ASSUMED = EETABLE
;                 (INTR ROUTINES USING DPTR SHOULD
;                 PUSH/POP).
;
;  DESTROYS:      ACC
;
;  RETURNS:       NEXT EET CONSTANT FROM LIST IN ACC
;
;  UPDATE HISTORY:
;
;*******************************************************

%*DEFINE(GET_EEC)(
$SAVE NOGEN
        INC     EET_OFF
        MOV     A,EET_OFF
        MOVC    A,@A+DPTR
$RESTORE
)
$EJECT
```

280640-50

Figure C-17. Macro GET_EEC

```
;*******************************************************
;
;  NAME:          VALERR
;
;  PURPOSE:       ROUTINE FOR REPORTING IC WRITE VALUE
;                 ERRORS
;
;  CALLED BY:     JUMPED TO FROM EDIT ROUTINES UPON
;                 VALUE ERROR DETECTION
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   CHECKS FOR LOCAL OR GLOBAL ACCESS AND
;                 UPDATES GENERAL STATUS REGISTER
;                 ACCORDINGLY WITH VALUE ERROR STATUS.
;                 RETURNS TO ICHDLR.
;
;  REG BANK:      ASSUMES 0, SELECTS NONE
;
;  INPUTS:        MPC_STS
;
;  DESTROYS:      NOTHING
;
;  RETURNS:       GENERAL STATUS REG, UPDATED FOR VALUE
;                 ERROR, TO ICHDLR
;
;  UPDATE HISTORY:
;
;*******************************************************

VALERR: JB    MPC_STS.PORT,LCVLER     ;IF LOCAL ACCESS
                                      ;THEN JUMP, ELSE
        ORL   GEN_STS,#GBL_VL_ERR  ;SET GENERAL STATUS
                                   ;= GLOBAL VALUE ERROR
        SJMP  ENDVAL                ;RESTORE AND RETURN
LCVLER: ORL   GEN_STS,#LCL_VL_ERR  ;SET GENERAL STATUS
                                   ;= LOCAL VALUE ERROR
ENDVAL: RET                        ;RETURN TO IC HANDLER

$EJECT
```

280640-51

Figure C-18. Report IC Write Value Error

```
;*******************************************************
;
;  NAME:          ER00H
;
;  PURPOSE:       GENERATE LOCAL STATIC ACCESS RIGHTS
;                 ERROR
;
;  CALLED BY:     JUMPED TO BY SERVICE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   EDIT VECTOR 0 GETS EDIT ENTRY 0,
;                 WHICH IS USED FOR LOCAL READ ONLY
;                 REGISTERS.  ONCE HERE, THE GENERAL
;                 STATUS REG IS UPDATED FOR LOCAL
;                 STATIC ERROR.  THE ACCESS IS THEN
;                 TERMINATED.
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        NOTHING
;
;  DESTROYS:      NOTHING
;
;  RETURNS:       NOTHING, RETURNS TO ICHDLR
;
;  UPDATE HISTORY:
;
;*******************************************************

ER00H:  ORL    GEN_STS,#LCL_ST_ERR          ;UPDATE GSR
        RET    ;BACK TO IC HANDLER TO TERMINATE ACCESS

$EJECT
```

280640-52

Figure C-19. ER00H Generate Local Static Access Rights Error

```
;*******************************************************
;
;  NAME:          ER01H
;
;  PURPOSE:       CHECK ALL RFU BITS = 0
;
;  CALLED BY:     JUMPED TO FROM SERVICE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   INDEXES INTO THE EDIT ENTRY TABLE TWO
;                 TIMES, ONCE FOR THE EDIT CONSTANT AND
;                 THEN A SECOND TIME FOR THE OFFSET TO
;                 THE WRITE ROUTINE.  ANDs CONSTANT
;                 WITH ICDATA.  IF RESULT > 0, THEN NOT
;                 ALL RFU = 0,  JUMPS TO VALUE ERROR
;                 ROUTINE, ELSE JUMPS INDIRECTLY TO
;                 WRITE ROUTINE.
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        EET_OFF,ICDATA
;
;  DESTROYS:      ACC, DPTR, EET_OFF
;
;  RETURNS:       NOTHING, JUMPS TO VALERR, ELSE WRITE
;                 ROUTINE
;
;  UPDATE HISTORY:
;
;*******************************************************

ER01H:  INC    EET_OFF           ;INC INDEX INTO EETABLE
        %LOOKUP(#EETABLE,EET_OFF)        ;GET CONSTANT
        ANL    A,ICDATA          ;CHECK VALUE OF ICDATA
        JNZ    ERR_1              ;JMP IF ILLEGAL VALUE
        %GET_EEC          ;GET WR ROUTINE OFFSET FM EET
        MOV    DPTR,#WRLBL    ;GET WR ROUTINE LABEL ADDR
        JMP    @A+DPTR                ;JMP TO WR ROUTINE
ERR_1:  AJMP   VALERR                  ;REPORT VALUE ERR
                                ;(TOO FAR FOR REL JMP)

$EJECT
```

280640-53

Figure C-20. ER01H Check for RFU Bits = 0

```
;*******************************************************
;
;  NAME:          ER02H
;
;  PURPOSE:       CONST_1 <= ICDATA <= CONST_2
;
;  CALLED BY:     JUMPED TO FROM SERVICE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   ICDATA BANDPASS FILTER.  IF VALUE OF
;                 ICDATA IS WITHIN THE RANGE SPECIFIED
;                 BY THE TWO CONSTANTS FROM THE EET,
;                 THEN THE WRITE ROUTINE OFFSET FROM
;                 EET IS JUMPED TO, ELSE A VALUE ERROR
;                 IS REPORTED IN THE GENERAL STATUS
;                 REG.  THUS EET MUST INCLUDE THREE
;                 BYTES:  CONST_1:  SMALLEST LEGAL
;                 VALUE FOR ICDATA, CONST_2:  LARGEST
;                 LEGAL VALUE FOR ICDATA, AND AN WRITE
;                 ROUTINE OFFSET
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        EET_OFF, ICDATA
;
;  DESTROYS:      ACC, DPTR, EET_OFF, TEMP
;
;  RETURNS:       NOTHING, JUMPS TO WRITE ROUTINE, ELSE
;                 VALERR
;
;  UPDATE HISTORY:
;
;*******************************************************
```

280640-54

Figure C-21. ER02H ICDATA between Two Values

```
ER02H:  INC    EET_OFF              ;EET_OFF = EET_OFF+1
        %LOOKUP(#EETABLE,EET_OFF)          ;GET CONST_1
        MOV    TEMP,A                    ;STORE CONST_1
        MOV    A,ICDATA          ;GET VALUE BEING WRITTEN
        %SUB(TEMP)                 ;SUB CONST_1 FM ICDATA
        JC     ERR_2            ;IF ICDATA < CONST_1 THEN
                                              ;VALUE ERR
        %GET_EEC                          ;GET CONST_2
        %SUB(ICDATA)               ;SUB ICDATA FM CONST_2
        JC     ERR_2            ;IF ICDATA > CONST_2 THEN
                                              ;VALUE ERR
        %GET_EEC                  ;GET WR ROUTINE OFFSET
        MOV    DPTR,#WRLBL     ;POINT TO WR ROUTINE LABEL
        JMP    @A+DPTR        ;JMP INDIRECT TO WR ROUTINE
ERR_2:  AJMP   VALERR           ;JMP TO VALUE ERR ROUTINE

$EJECT
```

280640-55

Figure C-21. ER02H ICDATA between Two Values

```
;*******************************************************
;
;  NAME:          ER03H
;
;  PURPOSE:        EDIT ROUTINE FOR PSB ARB ID REGISTER
;
;  CALLED BY:     JUMPED TO FROM SERVICE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   CHECKS ALL RFUs = 0, THEN CHECKS FOR
;                 LEGAL ARB ID BY COUNTING THE BIT
;                 TRANSITIONS IN ICDATA.
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        NONE
;
;  DESTROYS:      ACC, TEMP
;
;  RETURNS:       JUMPS DIRECTLY TO WR01H, ELSE VALERR
;
;  UPDATE HISTORY:
;
;*******************************************************

ER03H:  MOV   A,#PSB_ARB_MSK              ;GET RFU MASK
        ANL   A,ICDATA                  ;MASK ALL BUT RFUs
        JNZ   ERR_3     ;IF NOT ALL RFUs = 0 THEN ERROR
        MOV   TEMP,#TRANS_CNT           ;LOAD TRANS CNTR
        MOV   A,ICDATA          ;GET VALUE BEING WRITTEN
        CLR   C
        RLC   A                           ;SEE IF MSB SET
        JNC   ID_TR                 ;IF NO CARRY, THEN OK
        SETB  ACC.3                          ;ELSE SET BIT 3
                                  ;(ARB ID IS ONLY 5 BITS)
ID_TR:  XRL   A,ICDATA                ;ID BIT TRANSITIONS
ROT_A:  CLR   C                     ;ROTATE ZEROS INTO ACC
        RLC   A                     ;LOOK FOR TRANSITIONS
        JNC   CHK_ACC           ;IF CARRY BIT IS NOT SET
                                                ;THEN JMP
        DJNZ  TEMP,ROT_A ;ELSE TALLY UP ONE TRANSITION
ERR_3:  AJMP  VALERR        ;IF TRANS_CNTR REACHES ZERO
                                               ;THEN ERROR
CHK_ACC:JNZ   ROT_A                     ;IF ACC NOT ZERO
                                  ;THEN CONTINUE LOOKING
        AJMP  WR01H             ;ELSE JUMP TO WR ROUTINE

$EJECT
```

280640-56

Figure C-22. ER03H Edit PSB ARB ID Register

```
;*******************************************************
;
;  NAME:          ER04H
;
;  PURPOSE:       EDIT ROUTINE FOR PSB DIAG REGISTER
;
;  CALLED BY:     JUMPED TO FROM SERVICE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   CHECKS ALL RFUs = 0, THEN IF
;                 LOOPBACK MODE BEING SELECTED,
;                 CHECKS FOR RSTNC* ASSERTED.
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        NONE
;
;  DESTROYS:      ACC, DPTR
;
;  RETURNS:       JUMPS DIRECTLY TO WR06H, ELSE VALERR
;
;  UPDATE HISTORY:
;
;*******************************************************

ER04H:  MOV   A,#PSB_DIAG_MSK   ;GET PSB DIAG RFU MASK
        ANL   A,ICDATA              ;MASK ALL BUT RFUs
        JNZ   ERR_4               ;IF ALL RFUs NOT = 0
                                    ;THEN VALUE ERROR
        JNB   ICDATA.LBACK,END_E4     ;IF LOOPBACK NOT
                                ;SELECTED, SET THE WR REG
        MOV   A,#RST_TYPE_MSK       ;GET MASK FOR RESET
                                            ;TYPE BITS
        ANL   A,PSB_CS              ;MASK ALL BUT RESET
                                     ;TYPE FROM PSB C/S
        JZ    ERR_4         ;IF RSTNC HAS BEEN CLEARED
                                        ;THEN VALUE ERR
END_E4: AJMP  WR05H                           ;WR REG
ERR_4:  AJMP  VALERR              ;REPORT VALUE ERROR

$EJECT
```

280640-57

Figure C-23. ER04H Edit PSB Diagnostic Register

```
;*******************************************************
;
;  NAME:          ER05H
;
;  PURPOSE:       EDIT ROUTINE FOR REGISTERS
;                 REQUIRING > THAN CHECK
;
;  CALLED BY:     JUMPED TO FROM SERVICE
;
;  CALLS:         NONE
;
;  NEST LEVEL:    ??
;
;  DESCRIPTION:   CHECKS FOR ALL RFUs = 0.  MASKS OFF
;                 ANY DON'T CARE BITS, CHECKS FOR WRITE
;                 DATA (ICDATA) <= MAX ALLOWABLE VALUE.
;                 IF BOTH CHECKS PASS, THEN JUMPS
;                 INDIRECTLY TO WRITE ROUTINE, ELSE
;                 VALUE ERROR.  EET(1) = RFU MASK.
;                 EET(2) = DON'T CARE MASK.  EET(3) =
;                 MAX ALLOWABLE WRITE VALUE.  EET(4) =
;                 WRITE ROUTINE OFFSET.
;
;  REG BANK:      ASSUMES IC_BANK, SELECTS NONE
;
;  INPUTS:        ICDATA
;
;  DESTROYS:      ACC, TEMP, DPTR
;
;  RETURNS:       JUMPS INDIRECTLY TO WRITE ROUTINE,
;                 ELSE VALERR
;
;  UPDATE HISTORY:
;
;*******************************************************
```

280640-58

Figure C-24. ER05H Edit Register for Greater than Check

```
ER05H:  INC    EET_OFF     ;POINT TO RFU MASK IN EETABLE
        %LOOKUP(#EETABLE,EET_OFF)              ;GET IT
        ANL    A,ICDATA          ;CHECK FOR ALL RFUs = 0
        JNZ    ERR_5               ;JMP IF ILLEGAL VALUE
        %GET_EEC      ;GET DON'T CARE MASK FROM EETABLE
        ANL    A,ICDATA        ;MASK OFF DON'T CARE BITS
                                              ;FROM ICDATA
        MOV    TEMP,A                         ;SAVE RESULT
        %GET_EEC           ;GET MAX ALLOWABLE WR VALUE
        %SUB(TEMP)           ;SUB VALUE BEING WRITTEN
        JC     ERR_5  ;IF ICDATA > MAX ALLOWABLE VALUE,
                                             ;THEN ERR
        %GET_EEC           ;ELSE GET WR ROUTINE OFFSET
        MOV    DPTR,#WRLBL         ;POINT TO WR ROUTINE
        JMP    @A+DPTR               ;GO TO WR ROUTINE
ERR_5:  AJMP   VALERR              ;REPORT VALUE ERROR

$EJECT
```

280640-59

Figure C-25. ER05H Edit Register for Greater than Check

APPENDIX D
USER CALLABLE MACRO ROUTINES

```
;*******************************************************
;
;  MACRO NAME:    RD_MPC  (READ MPC)
;
;  SYNTAX:        RD_MPC(DEST,SRC)
;
;  PURPOSE:       READ REGISTER FROM MPC IC REGISTER
;                 SET
;
;  DESCRIPTION:   THE VALUE CONTAINED IN OR BY SRC (MAY
;                 BE CONSTANT OR DIRECT ADDRESS) MUST
;                 BE A LEGAL MPC REGISTER ADDRESS.  THE
;                 REGISTER DATA IS RETURNED IN THE
;                 LOCATION SPECIFIED IN DEST.
;
;  INPUTS:        DEST, SRC
;
;  DESTROYS:      MPC_RNUM, ACC
;
;  RETURNS:       MPC REGISTER DATA IN DEST
;
;  UPDATE HISTORY:
;
;*******************************************************

%*DEFINE(RD_MPC(DEST,SRC))(
$SAVE nogen
        MOV     MPC_RNUM,%SRC
        MOVX    A,@MPC_RNUM
        %IF(%NES(A,%DEST)) THEN (
        MOV     %DEST,A
        )FI
$RESTORE
)

$EJECT
```

280640-60

Figure D-1. Macro RD__MPC

5

```
;*******************************************************
;
;  MACRO NAME:    WRT_MPC  (WRITE MPC)
;
;  SYNTAX:        WRT_MPC(DEST,SRC)
;
;  PURPOSE:       WRITE DATA TO MPC IC REGISTER
;
;  DESCRIPTION:   THE VALUE CONTAINED IN OR BY SRC (MAY
;                 BE CONSTANT OR DIRECT ADDRESS) IS
;                 WRITTEN TO THE MPC REGISTER SPECIFIED
;                 BY DEST, WHICH MAY ALSO BE A CONSTANT
;                 OR DIRECT ADDRESS.
;
;  INPUTS:        DEST, SRC
;
;  DESTROYS:      MPC_RNUM, ACC
;
;  RETURNS:       NOTHING
;
;  UPDATE HISTORY:
;
;*******************************************************

%*DEFINE(WRT_MPC(DEST,SRC))(
$SAVE nogen
        MOV     MPC_RNUM,%DEST
        %IF(%NES(A,%SRC)) THEN (
        MOV     A,%SRC
        )FI
        MOVX    @MPC_RNUM,A
$RESTORE
)

$EJECT
```

280640-61

Figure D-2. Macro WRT__MPC

```
;*******************************************************
;
;  MACRO NAME:    LOOKUP
;
;  SYNTAX:        LOOKUP(TABLE,OFFSET)
;
;  PURPOSE:       RETRIEVE BYTE FROM TABLE.
;
;  DESCRIPTION:   THE SPECIFIED TABLE IS INDEXED BY THE
;                 VALUE OF OFFSET AND RESULTING ADDRESS
;                 IS READ INTO THE ACCUMULATOR.
;
;  INPUTS:        TABLE, OFFSET
;
;  DESTROYS:      DPTR, ACC
;
;  RETURNS:       DESIRED BYTE IN ACC
;
;  UPDATE HISTORY:
;
;*******************************************************

%*DEFINE(LOOKUP(TABLE,OFFSET))(
$SAVE nogen
        %IF(%NES(A,%OFFSET)) THEN (
        MOV     A,%OFFSET
        )FI
        MOV     DPTR,%TABLE
        MOVC    A,@A+DPTR
$RESTORE
)

$EJECT
```

280640-62

Figure D-3. Macro LOOKUP

```
;*******************************************************
;
;  MACRO NAME:    SUB     (SUBTRACT)
;
;  SYNTAX:        SUB(OPR)
;
;  PURPOSE:       SUBTRACT OPERAND FROM ACC
;
;  DESCRIPTION:   THE CARRY FLAG IS CLEARD AND THE
;                 VALUE SPECIFIED IN OR BY OPR IS
;                 SUBTRACTED FROM THE ACCUMULATOR.
;
;  INPUTS:        OPR
;
;  DESTROYS:      ACC, CARRY FLAG
;
;  RETURNS:       RESULT IN C AND ACC
;
;  UPDATE HISTORY:
;
;*******************************************************

%*DEFINE(SUB(OPR))(
$SAVE nogen
        CLR     C
        SUBB    A,%OPR
$RESTORE
)

$EJECT
```

280640-63

Figure D-4. Macro SUB (Subtract)

```
;*******************************************************
;
;  MACRO NAME:    MOVBIT  (MOVE BIT)
;
;  SYNTAX:        MOVBIT(DEST_BIT,SRC_BIT)
;
;  PURPOSE:       PERFORM A BIT MOVE OPERATION
;
;  DESCRIPTION:   THE SOURCE BIT IS MOVED TO THE CARRY
;                 FLAG.  THE CARRY FLAG IS MOVED TO THE
;                 DESTINATION BIT.
;
;  INPUTS:        DEST_BIT, SRC_BIT
;
;  DESTROYS:      CARRY FLAG
;
;  RETURNS:       DEST_BIT = SRC_BIT
;
;  UPDATE HISTORY:
;
;*******************************************************

%*DEFINE(MOVBIT(DEST_BIT,SRC_BIT))(
$SAVE nogen
        MOV     C,%SRC_BIT
        MOV     %DEST_BIT,C
$RESTORE
)

$EJECT
```

280640-64

Figure D-5. Macro MOVBIT

5

```
;*******************************************************
;
;  MACRO NAME:    GET_EEC (GET EDIT ENTRY CONSTANT)
;
;  SYNTAX:        GET_EEC
;
;  PURPOSE:       INDEX INTO EETABLE TO GET NEXT EDIT
;                 ENTRY CONSTANT
;
;  DESCRIPTION:   EET_OFF IS INCREMENTED AND THE CODE
;                 BYTE AT EET_OFF FROM DPTR (ASSUMED TO
;                 BE POINTING TO EETABLE) IS MOVED TO
;                 THE ACCUMULATOR.
;
;  INPUTS:        NONE, HOWEVER DPTR ASSUMED = EETABLE
;                 (INTR ROUTINES USING DPTR SHOULD
;                 PUSH/POP).
;
;  DESTROYS:      ACC
;
;  RETURNS:       NEXT EET CONSTANT FROM LIST IN ACC
;
;  UPDATE HISTORY:
;
;*******************************************************

%*DEFINE(GET_EEC)(
$SAVE NOGEN
        INC     EET_OFF
        MOV     A,EET_OFF
        MOVC    A,@A+DPTR
$RESTORE
)
$EJECT
```

280640-65

Figure D-6. Macro GET__EEC

intel®

AP-431

APPLICATION NOTE

Multibus II Simplifies Partitioning of a Complex Design

5

FRANCOIS HUGUENIN
SENIOR FIELD APPLICATION ENGINEER
INTEL SWITZERLAND

August 1993

Order Number: 280691-001

Multibus II Simplifies Partitioning of a Complex Design

CONTENTS

PAGE

PREFACE 5-127

1.0 INTRODUCTION 5-128

2.0 DESIGN PROBLEM OF THE SIMULATION OF THE JET ENGINE 5-128

2.1 The Engine Model 5-128

2.2 The Design Problem 5-130

2.3 The Uni-Processor Approach 5-130

2.4 The Multiprocessor Approach 5-131

3.0 MULTIPROCESSOR SOFTWARE DESIGN WITH iRMX II 5-132

3.1 The I/O Subsystem 5-133

3.2 Choosing a Communications Protocol 5-133

3.3 The Final Configuration and State Cycle 5-134

3.4 The Cycle Time Analysis 5-134

3.5 The iRMX II Host 5-135

4.0 THE APPLICATION PROCESSOR DESIGN 5-137

5.0 DEBUGGING AND PERFORMANCE ANALYSIS 5-137

6.0 SUMMARY AND CONCLUSIONS .. 5-139

CONTENTS

PAGE

APPENDICES

A. Bibliography and Related Readings .. 5-140

B. Glossary of Terms 5-141

C. Documented Portions of the iRMX II Host Monitor Program 5-142

D. Documented Code Listing of the Application Software Based in EPROM 5-150

E-1. I2ICE 286 Debug Session Sample .. 5-162

E-2. iPAT Analysis Protocol 5-167

LIST OF FIGURES

Figure 2-1. The CF6-50C Engine Layout 5-129

Figure 2-2. Engine Model Structure for Steady State Operation 5-129

Figure 2-3. Example of the Fuel Subsystem Model 5-130

Figure 2-4. The Engine Model Architecture for the DC-10 5-130

Figure 2-5. Functional Partitioning Using Four Processors 5-131

Figure 3-1. Basic Architecture of the System 5-132

Figure 3-2. The Application System Architecture 5-133

Figure 3-3. The Processor Activity Timing Diagram 5-134

Figure 3-4. The iRMX II Comm-Layer Unsolicited Message Format 5-135

Figure 3-5. Data Processing on the MPC_DATA Structure 5-136

Figure 5-1. The Application Processor Timing Analysis 5-138

PREFACE

Several years ago, prior to joining Intel, I worked at the Swiss Federal Institute of Technology in the Group for Automatic Control. As part of my job responsibilities there, we made some real-time simulation studies in the field of aircraft performance computers. The engine model was implemented on an 8086-based single board computer. It also had special custom-made hardware for displaying the instruments similar to those in the original aircraft cockpit. During the course of this work, we were able to demonstrate the feasibility of using microcomputers in this application field. A paper describing this work was then presented at the AIAA Flight Simulation Technologies Conference, Long Beach, California in June 1980.[1] While the use of microcomputers in aircraft simulation has become quite common in recent years, this work anticipated the impact of major cost reductions and performance improvements which were to follow in the flight simulation industry.

In more recent years, I have served as an Application Engineer for Intel Corporation in the Swiss District. In this role I am often confronted with the problem of how to demonstrate the capabilities of a product in a way which engineers can easily relate to and understand. This was the case with Multibus II—an industry standard bus, specially designed for the more rigorous demands of multiprocessing. Since few applications today are partitioned to take advantage of parallel computing, my goal was to demonstrate how functional partitioning could be applied to a computationally intensive application with relative ease. The goal of breaking the "Von Neuman bottleneck" would be to increase the aggregate computing power without a substantial increase in overhead.

For assistance on this project, I called upon the Furrer & Gloor Company, a Multibus manufacturer with broad experience in industrial automation. With their assistance I was able to modify the work done seven years ago on aircraft simulation and convert it to Multibus II to demonstrate multiprocessing. This application provides a good example of the typical Multibus II design cycle. We begin with the basic architectural decisions, define the message contents between processors, show the use of software development and debug tools, and then finally test for performance tuning on the final system once it is up and running. As you read through this document you will learn about the process of application development in addition to the special demands of the aircraft simulation experiment.

I would like to thank Markus Schoenbucher and Christoph Graf of the Furrer & Gloor Company for their great support during the weeks of implementation in the labs. George Walker of WeDV deserves special thanks for designing and debugging the iRMX®II part of the application. I wish also to thank P. Marti, H. -R. Aeschilmann, K. Krizaj and B. Leiser of the Simulator Maintenance Group at Swissair in Zurich who gave me access in the early phase to all the important engine data and most valuable inputs on the flight simulation technology in general back in the late 70's. Finally I especially want to thank Roger Finger from Multibus II Application Engineering at the Intel factory in Hillsboro, Oregon for his guidance in helping me get this Ap Note done and for his corrections to my "Swiss English".

—Francois Huguenin

1.0 INTRODUCTION

Aircraft simulation trains pilots at a substantial savings. Flight crews can train around the clock, without regard to weather conditions and airport congestion. This important simulation model is a complex technical system which includes several high-performance digital computers. One key part of the simulator is the engine model: it is vital to the functioning of the entire machine:

— For the flight model equations (thrust and torque)

— For the "man in the loop" process with the important flight deck instruments (the pilot reacts on the settings and feedback from the engine instruments)

— For the flight engineer with all subsystems which are fed by the engines (hydraulics, electrical power supply, air conditioning, anti-icing, etc.)

The basic problem for good simulation is computing speed, because the quality of the pilot's training is a direct function of the flight simulator's update speed. Previously most simulators were using more than one processor based on a shared memory architecture. The trend today is dictated by the ever-increasing complexity of modern aircraft which are using more and more electronics on board. The new generation of flight simulators has a collection of black boxes (navigation computer, display units, engine control) used directly as in the original aircraft. In this complex environment, the engine is a closed subsystem which can be modeled without having to build a whole flight simulator. This part of the machine has well-known interaction points to the rest of the simulator. The work which was done at the Swiss Federal Institute of Technology some years ago was to be part of an on-board performance computer to calculate fuel optimal flight trajectories.[2] The following implementation of the engine model with Multibus II is based on these results.

2.0 DESIGN PROBLEM OF THE SIMULATION OF THE JET ENGINE

The General Electric CF6-50C Fan Jet Engine which is installed on all of Swissair's DC-10 aircraft provides our study model. This engine of the now "older" generation delivers up to 75% of the thrust power through the fan. The model designed during the years 1978 to 1980 is based on data made available by the engine manufacturer General Electric, McDonnell Douglas for the installed data (measurements made during the certification of the DC-10) and finally from CAE, the flight simulator manufacturer, with the actual programs written in assembler for the SIGMA computer.

2.1 The Engine Model

In order to be as realistic as possible, the engine model is very complex and has the following main characteristics:

— It is multivariable, with secondary interdependencies.

— It has basically two states: the transient state when coming up after ignition, and steady state when being held at a stable working point.

— It is a function of many parameters which directly influence the engine.

A cross section of the CF6-50C engine is shown in Figure 2-1.

Undoubtedly, the main problem of the model is the fuel control unit, which controls the whole engine. This unit also defines the transient behavior of the engine startup and excursions of the working point in the steady state mode. Of course, some simplifications are necessary for simulation. It was necessary to concentrate on the steady state model to reduce the modeling problem to a manageable task.[1] No startup sequence of the engine was to be modeled, but this approach has the great advantage of allowing the direct use of the previously mentioned data tables from the engine manufacturer. The model was exact in the range for which the data tables were originally measured. This reduced and more manageable model is shown in Figure 2-2.

A collection of modules are building the model with some interactions between them. These subsystems are:

— INLET:	Reduction of mach, temperature and pressure to standard values sea level, standard day and temperature
— POWER LEVEL:	Model of the power lever in the cockpit. From these, the pilot sets the throttles
— N2 TRANSIENT:	Simplified model of the fuel control unit (basically a fourth order digital filter)
— N1:	Model of the FAN as function of mach and N2
— THRUST:	Model of the thrust as function of N1 and N2
— FUEL FLOW:	Model of the fuel flow inside the engine in tons per hour as function of N2, altitude and mach
— EGT:	Model of the engine temperature as function of N2 and mach
— EPR:	Model of the engine pressure ration as function of mach and N2

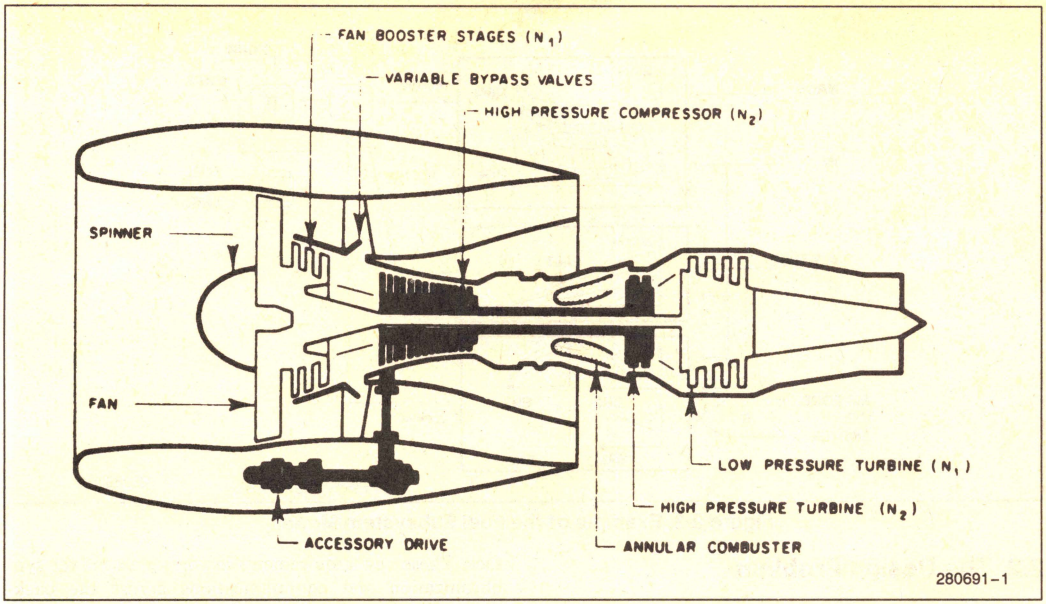


Figure 2-1. The CF6-50C Engine Layout

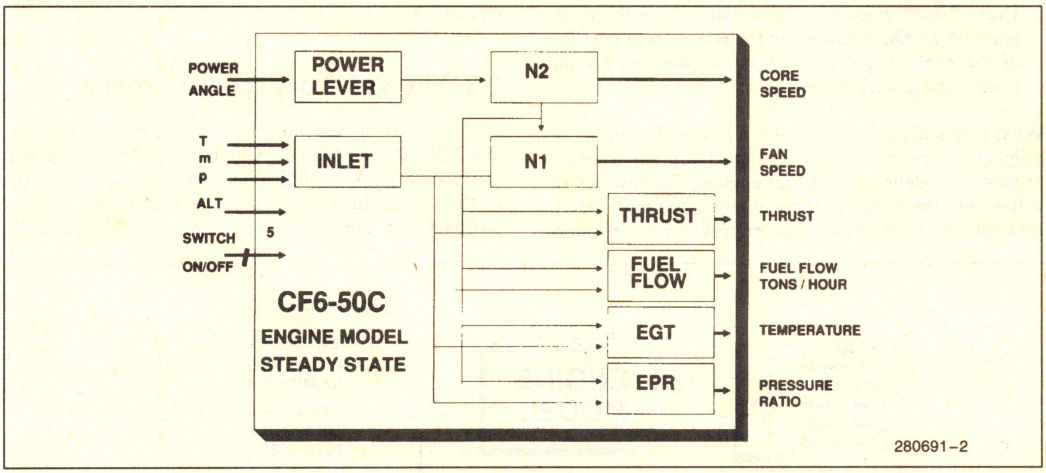


Figure 2-2. Engine Model Structure for Steady State Operation

The power plant of an aircraft is, as mentioned before, a closed system and provides additional tasks besides thrust for motion. The engine must provide fresh air for the cabin, and electric power for the cockpit, kitchen, cockpit electronics, etc. It must also activate and sustain the hydraulic subsystem for the control surfaces of the wings, and feed the anti-icing system for the wing tips, etc. The model, to be realistic, must take into account that these additional tasks will also have an appreciable influence on the state of the engine.

This can be demonstrated best with the fuel model as shown in Figure 2-3. The fuel model is basically a function of mach and N2 (referred core speed). This gives the main data curve and the basic fuel flow in tons per hour and will be interpolated through a high-speed algorithm. When the aircraft is climbing, the fuel flow will change according to a second data curve for correction with the value DELTA EWFA1. The correction, due to the on or off switching of air conditioning, etc., will also be taken into account with other data tables. The total fuel flow, as well as the fuel used by the engine, finally can be calculated with all the deltas. This model was validated with actual data taken from the Aircraft Data System of the DC-10.[1]

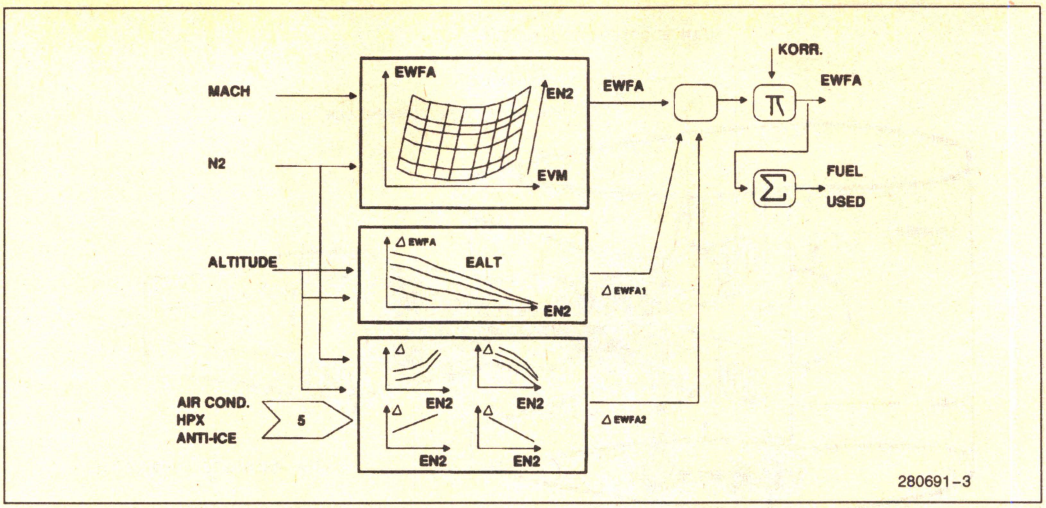


Figure 2-3. Example of the Fuel Subsystem Model

2.2 The Design Problem

The task to be done now can be summarized as follows:

> **Every 20 ms calculate a new state of all three engines of the DC-10 based on the power lever position of the cockpit and the subsystem switches for air conditioning, anti-ice, etc.**

What seems trival in one sentence is in reality not! The original engine model was designed around a single processor modeling a single jet engine. The new design will extend the model to three engines, and will add an I/O subsystem to provide an improved human interface. These new requirements introduce a need for synchronization and communications across the backplane. The implementation of the single processor system will be discussed first, followed by the multiprocessing extensions.

2.3 The Uni-Processor Approach

The original engine model was designed around an iSBC 86/12 Single Board Computer (without numeric coprocessor).[1] A small display unit was built as a copy of the central instrument panel of the cockpit to give feedback on the model's status (see Figure 2-4).

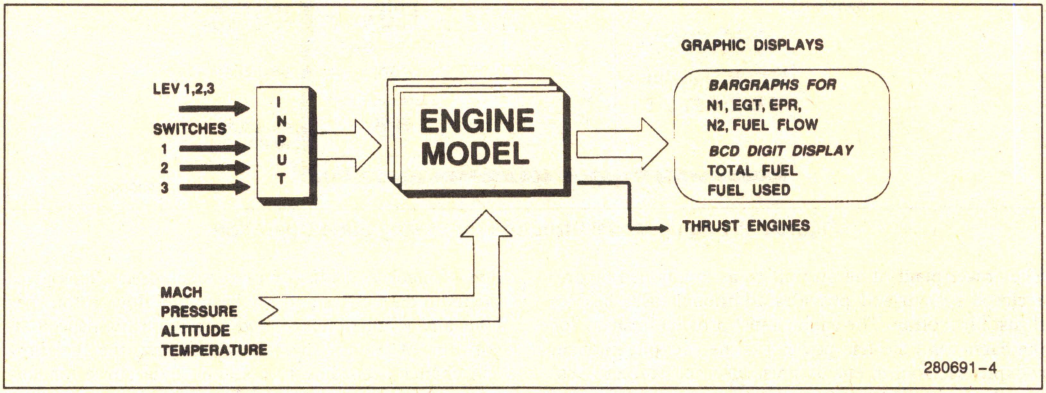


Figure 2-4. The Engine Model Architecture for the DC-10

With a single processor, the software architecture is relatively simple: on every clock interrupt, the I/O unit was activated to read input data (power level position, on/off switches, etc.) and the engine model called three times in sequence with the appropriate parameters. Afterwards, some scaling took place to display the bar graphs and fuel used digits (see Figure 2-4). Using this approach, the compute time depended on programming quality in the loop and therefore was optimized using assembler coded routines. The complete computing cycle consists of: read the inputs, calculate the engine model three times with the new state due to input parameters, scale the results and output them to the I/O unit.

2.4 The Multiprocessor Approach

To relieve the computational bottleneck and further reduce the update time, additional processors are needed. The basic question when using multiple processors is how to partition the application (in this case the modeling of all three DC-10 engines). For this application, one processor for each engine is appropriate because each engine model can be considered as a closed system with well-known interaction points. This results in a four-computer system: one processor must be a kind of "master" to coordinate the I/O and simulate the flight simulator; the remaining three are each modeling one engine of the DC-10.

The data flow changes slightly compared to the one processor approach. Because each processor is independent, each must be synchronized in some way.

Using the Multibus II technology[4] overcomes the hurdles of designing a multiprocessing hardware and is available off the shelf. The synchronization uses messages, exchanged between the host and its partners allowing a data exchange. This results in the architecture shown in Figure 2-5 where CPU #2, #3, #4 all have a local implementation of the engine model. The host or CPU #1 will handle all the coordination work and some key functions for the transient model and display scaling. The coupler board MBII/LNK-1 will handle the input-output processing.

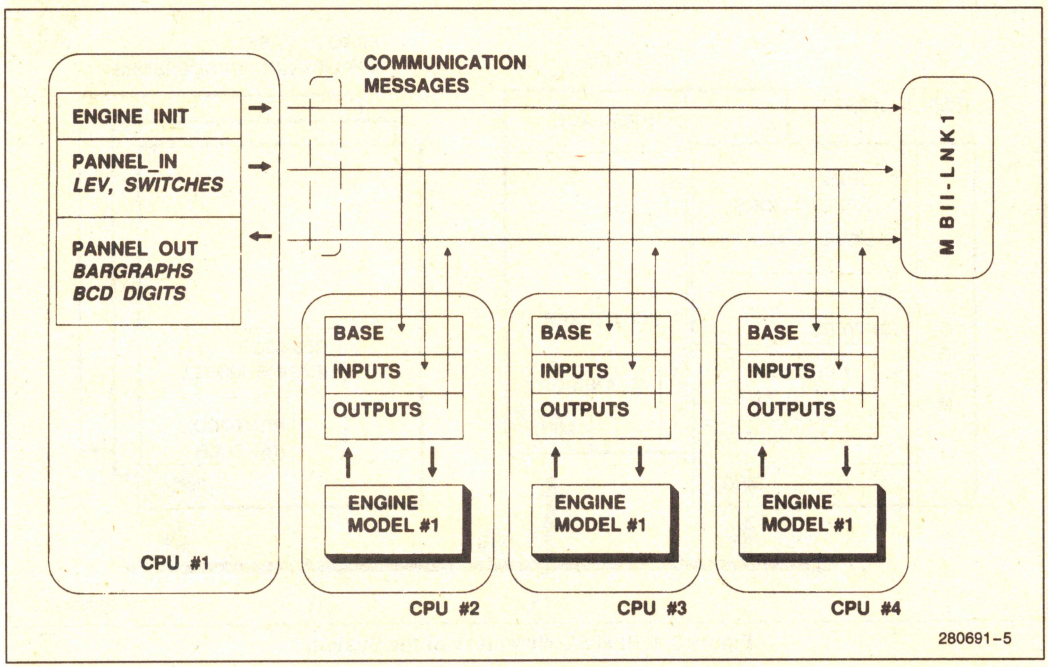


Figure 2-5. Functional Partitioning Using Four Processors

3.0 MULTIPROCESSOR SOFTWARE DESIGN WITH iRMX® II

The architecture chosen for the application is based on a fully-configured iRMX II system as a host, complemented with three "EPROM"-based application processors, one for each engine of the DC-10. This decision was made to minimize the amount of recoding which would be necessary for the original engine model. Software which was written a couple of years ago can be reused without change. In the process, the older iSBC 86/12 hardware will be upgraded to iSBC 286/100A boards for faster execution speed and Multibus II compatibility. Figure 3-1 shows the system architecture of the new multiprocessing system.

Each of the four processors in this system has some primary task to fulfill:

Host:

This is the central control processor which runs a complete real-time operating system. The iRMX II operating system was configured with a human interface, disk I/O subsystem (PCI—Peripheral Communications Interface), and a message passing communications layer. The iRMX II console will also be used as an on-target host for software development.

Applications Processors:

Each of the applications processors runs an EPROM based program which contains the engine model software. The only change which was required to the original software was the addition of a message passing communications interface. Since there is no operating system on these boards, a low-level MPC protocol handler is responsible for this function.

LNK-1:

This board serves as a bridge to the low-cost I/O subsystem and the master clock during simulation time.

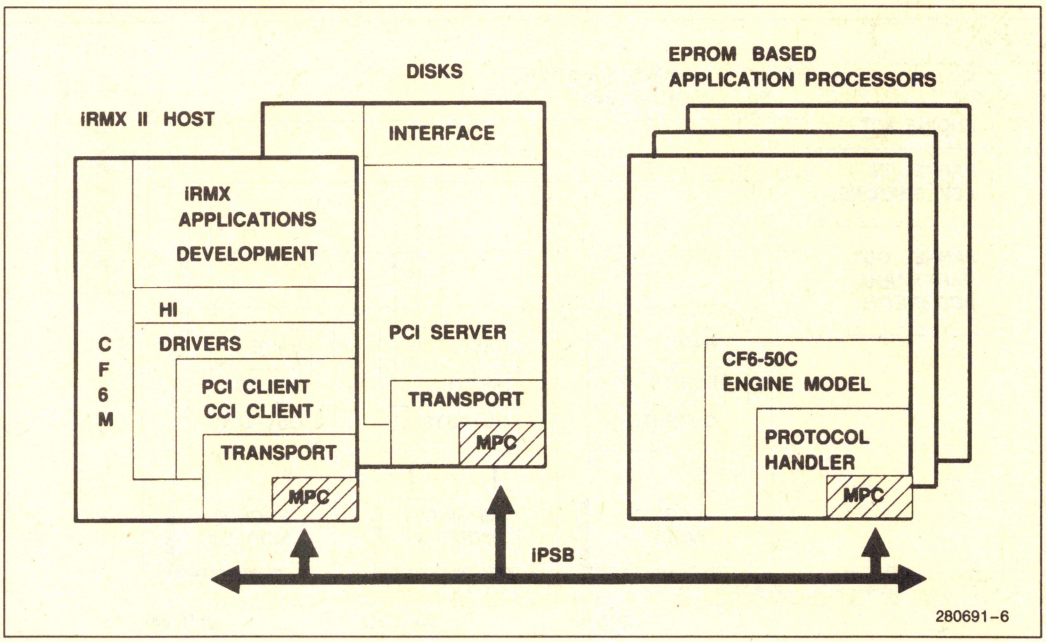


Figure 3-1. Basic Architecture of the System

3.1 The I/O Subsystem

As previously mentioned, the engine model also contains a display unit for control of the model. In order to add this important function, a coupler board manufactured by an independent MMG vendor (Multibus Manufacturers Group) was used as a bridge to the I/O subsystem, as shown in Figure 3-2. The I/O system interfaces to:

— A panel with the "on/off" switches for each engine (anti-ice subsystem, etc.), as mentioned in Section 2.1.

— An A/D converter to read and convert the position of the throttles to a binary value.

— A video controller to display the state of the engine as on the cockpit instruments.

In the following sections we will discuss in detail the various steps necessary to initegrate the above design into a real Multibus II system:

— Choice of the messages and their structure

— HOST startup sequence, iRMX II related topics

— Use of the iRMX II communications layer

— "Application Processor" design

3.2 Choosing a Communications Protocol

The Multibus II architecture defines several types of processor access to the parallel system bus:[4.1] solicited message transfer, unsolicited message transfer, dual port memory access and interconnect address access. Interconnect address space will be used for configuration and initialization of the Multibus II system and will not be accessed following system start-up. Dual-port memory as implemented on the MBII/LNK-1 board is used in the design[3] and allows use of unsolicited messages and memory transfer. The transfer of information between the host and the application processors is expected to require high speed transfers of numerous small blocks of data. For this, message passing is ideal.

The decision as to what type of message transfer will be used (solicited or unsolicited) will be based on how much data has to be transferred among the processors. As shown in Figure 2-5, there are basically three types of messages to be transferred:

—BASE: This message contains the base information, such as mach, pressure and altitude for the INLET portion of the engine model (Figure 2-2). This message is outbound from the host only.

—INPUTS: This message contains the N2 variable (core speed) of the engine, as calculated from the preprocessing in the Host. It also contains the decoded switch information from the panel of the model "cockpit" for the "on/off" switches of the subsystems, i.e., anti-ice, etc.

—OUTPUTS: This message has much more data to transfer back to the host for the final processing: all main computed data from the model must be sent back to the host.

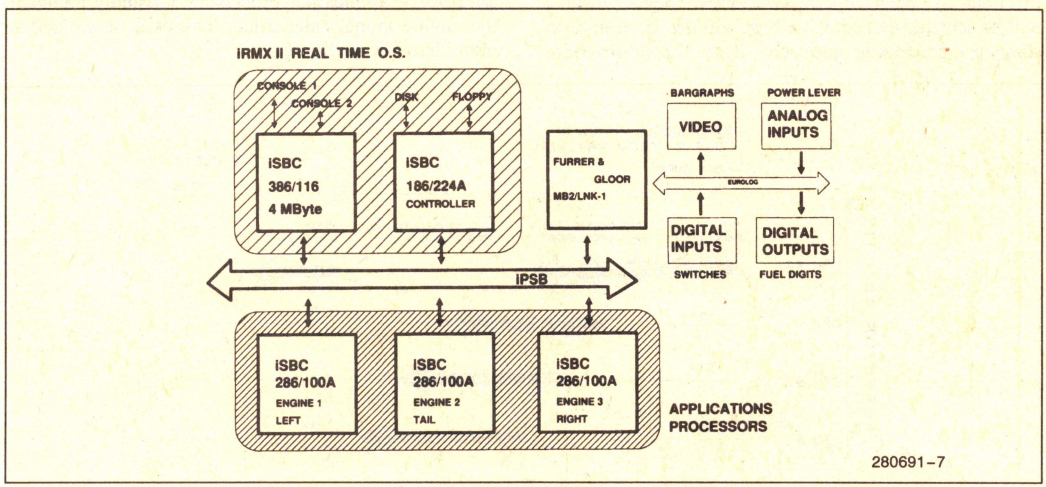


Figure 3-2. The Application System Architecture

Once the structure of "what has to be transferred" is defined, detailed analysis begins on the amount of data to be transferred. In this application the number of bytes to be transferred is relatively small, as follows: BASE—3 words; INPUTS—7 words; OUTPUTS—9 words. During the implementation phase it was quickly realized that it would be wise to define a flag word to ease the debugging task. This flag word is an overhead, but it allows identification of messages being received for debugging.

The next choice to make is what kind of message: unsolicited or solicited? Using solicited messages means that a certain amount of data has to be sent between the processors in order to be useful. In this design, the use of a solicited message will mean only overhead and complexity in the software. Only a few words need to be transferred. Therefore the choice is to simplify and use unsolicited messages only. In a case where a data transfer with more than 20 bytes is needed, an unsolicited message may be sent twice. This is faster than setting up an entire solicited transaction.[4.3] The big advantage is simplification of application software and of the debugging task.

3.3 The Final Configuration and State Cycle

Looking at the final configuration of the system as shown in Figure 3-2, there will be a number of transfers through the PSB for each simulation cycle. Each of them will be started by the local processor sitting on the EUROLOG local bus. This processor will read the analog data, convert it to a binary value, read the switch positions and pack it all in a Multibus II message. This will be sent to the host. The host will decode it and do some pre-processing and send it to the appropriate application processor (or engine). After a simulation step has been done locally, each of the application processors will send the result back to the host for encoding the display and some post-processing. Afterwards, the results will be sent back to the MBII/LNK-1 board using two unsolicited messages for the display process because there are more than 20 bytes needed for this task. Therefore, a complete simulation cycle will involve nine unsolicited transfers among the various processors.

3.4 The Cycle Time Analysis

The next task is to look at the activities of all processors with respect to time. For this purpose, a "processor activity timing" diagram, as shown in Figure 3-3, is set up. The horizontal axis is time; the vertical has one entry for each processor. During the work cycle, in this case one simulation step based on the incoming unsolicited message from the MBII/LNK-1 coupler board, the main tasks of each of the processors is estimated and shown with an arrow. Each outgoing or incoming message is a transition vertical arrow to the next processor and means the start or the end of an activity.

Using this diagram, it is now possible to make a quantitative analysis about the workload of each of the processors. In this case we see that the CPU #2, #3, and #4 are usually sitting idle waiting for messages. This simply means that there is a lot of spare processing time available. The diagram also shows that during some time in the cycle there is true parallel processing happening, which confirms the design goal of shortening the simulation calculation cycle. The compute time of each of the application processors is usually bound to the engine model calculation. This will be studied in more detail in Section 5.

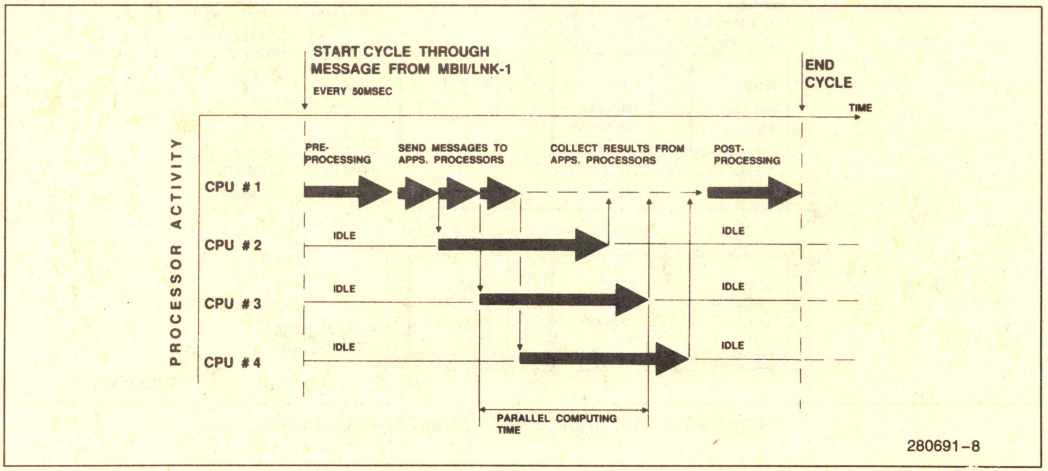


Figure 3-3. The Processor Activity Timing Diagram

3.5 The iRMX® II Host

As earlier mentioned, there will be a human interface (HI) job running under iRMX II which will monitor all the simulation activities. This monitor program will use the comm layer of iRMX II and be responsible for the startup of the system.

One part of the monitor program will also be the pre- and post-processing for the engine model as mentioned before. The startup sequence will include a board scan in order to be able to locate the application processors and define the necessary communications ports. The monitor program will function as follows:

— Create the connections to the terminal

— Scan the backplane and check the BIST (Built-In Self Test) status of each board

— If okay, then define the ports and sockets for the iRMX communications layer

— Create a buffer pool to preallocate free memory segments

— Create the iRMX memory segments and release them to the buffer pool

— Attach pool to port

— Ask for an amount of fuel in tons at the console

— Send the startup synchronization message to CPU #2, #3, #4 and to the MBII/LNK-1 board

— Send the start command to the MBII/LNK-1 board and give it control

— Do every 50 ms until no fuel available.

  — Receive the message from the MBII/LNK-1 board with the decoded power lever angle and switches

  — Make the preprocessing

  — Send to each application processor the incoming data for a simulation step

  — Collect the resulting messages asynchronously (there might be a slower board . . . !)

  — After having received all the results, make the post-processing

  — Send the two result messages back to the MBII/LNK-1 board. This finishes the "new state calculation" cycle.

— When no more fuel, stop everything and ask again at the console for more fuel

The coding of this monitor program is done in PL/M 286 using the iRMX II system calls. All the features of the PL/M 286 are used, especially the STRUCTURE DEFINITIONS which are very useful in this kind of application programming. Portions of the code are shown in Appendix B. Note that the transport system calls are very easy to use.

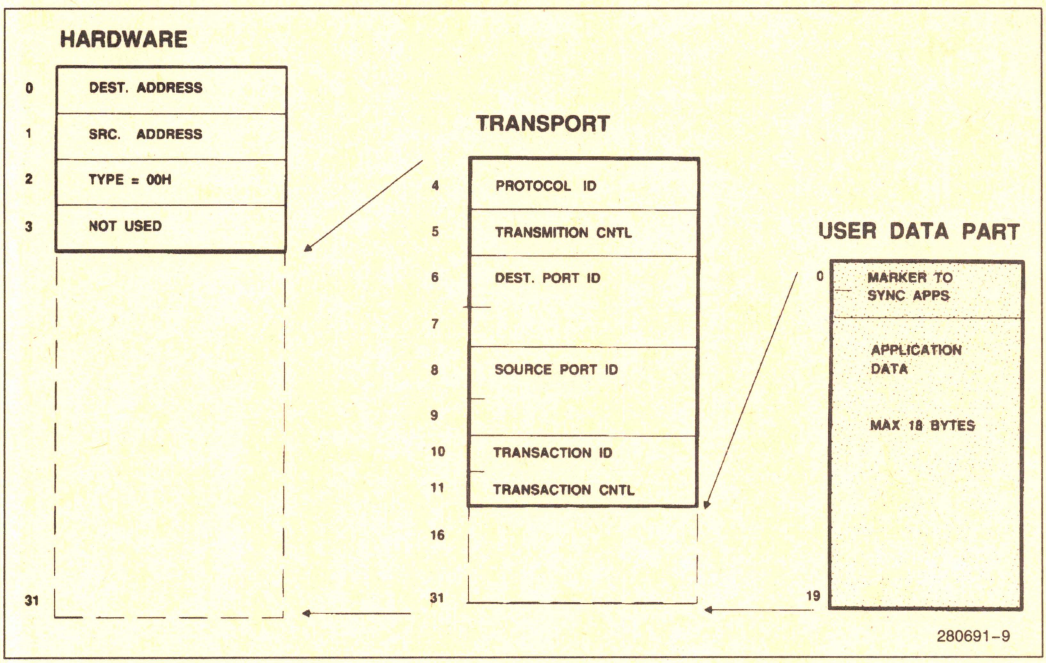


Figure 3-4. The iRMX II Comm-Layer Unsolicited Message Format

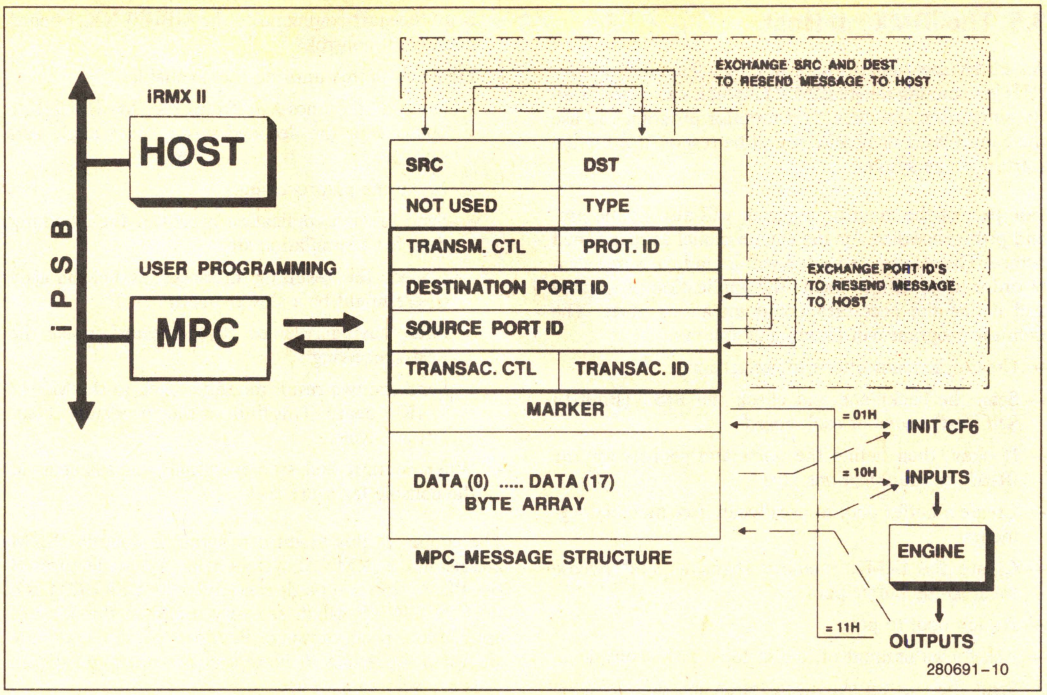


Figure 3-5. Data Processing on the MPC__DATA Structure

4.0 THE APPLICATION PROCESSOR DESIGN

The structure of the local software is, in principle, simple, due to the task which must be performed and can be split in two portions:

— A main program which will initialize the board and the whole local software, and

— An interrupt handler to handle the incoming messages accordingly. In addition, the interrupt handler calculates the new state of the engine and contains the engine model which needs no modification.

This leads to the design of several separate modules with well-known tasks:

—MAIN:	Init of the board upon cold reset, (int. controller, MPC)
—INTERRUPT HANDLER:	Activated on an MPC hardware interrupt. Will initialize the engine model or calculate the next state of the engine (so-called simulation step).
—MPC-LOW LEVEL DRIVER:	Receive a message, transmit a message, init the MPC, error case.

The MPC low level driver routines can be directly taken from the MPC User's Manual.[4.2] The programming was done using 32 byte messages since the structure of the application fits exactly in the unsolicited message format. However there are some implications when receiving an iRMX II message sent under the comm layer. The general format of the message is shown in Figure 3.4. This explains that not all available bytes can be used in the unsolicited message transfer since eight of them are used for the transport protocol overhead, making the logical task binding over the bus possible.[4.3] Therefore, the user has to be careful when using the message formatted under iRMX II and received locally without any operating system software. Since only one single task runs on the board (our interrupt handler), no port and sockets are needed locally, but the host has a port and a socket defined for accessing the application processor. To transmit the data back as described in the design section above, the comm layer must "understand" what is coming back. The following operations ensure this:

1. The source and destination fields of the message structure will have to be exchanged before returning the message with the engine simulation results. That is for the low-level hardware and is absolutely logical.
2. Exactly the same has to be done for the transport part of the message structure. The DESTINATION_PORT_ID and the SOURCE_PORT_ID have to be exchanged to allow the comm layer software to receive the results correctly.

The second step seems trivial, but it took several hours in front of the emulator to understand why the engine variable N1, a result of the simulation step, was always the same value. We had not taken into account the unsolicited message set up under iRMX II which has the whole TRANSPORT PROTOCOL integrated. Once this was discovered, we had no problems at all. Therefore, all applications using no operating system, kernel or executive locally, must have an "own" protocol handler implemented. Main portions of the documented code listings for total implementation of the application processor can be found in Appendix C.

The "data processing" done flag in the MPC_MESSAGE data structure is shown in Figure 3-5. The previously mentioned marker differentiates the kind of message and is, in some sense, part of a "user application protocol". In this case it allows initialization of the engine software or to make a simulation step.

5.0 DEBUGGING AND PERFORMANCE ANALYSIS

The debugging of the application for the local processor was done using the I2ICE 286 emulator. This instrument allows a very efficient debugging, especially when using the macro facility to display data. Therefore a small library of macros was written to display the incoming and outgoing messages. With this it was possible to quickly locate the above mentioned problem of the missing TRANSPORT adaptation. The whole applicaton was debugged in the high speed RAM of the emulator and afterwards relocated to be EPROM resident. However, debugging a multiprocessor system is, by definition, not simple because of all the coordination involved. The debug session around the application processor was based on the following methodology:

— Send a known message from the host

— Debug the MPC_RECEIVE_MESSAGE routine step by step and get the known message to local RAM

— Loopback with same message to check the MPC_TRANSMIT_MESSAGE routine

— When the loopback is okay, then add the full engine calculation and . . . it runs!

A small log of a debug session with I2ICE 286 and the use of the predefined macros can be found in Appendix D. It shows the big advantage of using symbolic debugging.

Afterwards, an analysis with the iPAT Performance Analyzer was done to get more confidence about the timing situation. Most impressive was the terrific speed of the whole thing.

The iPAT performance analyzer offers may features such as profile, coverage, usage, linkage mode, etc.[5] In this kind of environment, the objective was to measure how long the various elements of the cycle are, so only the duration mode was of read use. Of first interest was to check how much time is spent in a routine and the latency from the MPC Interrupt Hardware signal to the start of a particular routine. Initially the cycle time of 50 ms was chosen to have enough spare time due to the use of the relatively slow 5 MHz iSBC 86/12A. A time of about 15.5 ms could be achieved with the Multibus I system.

Porting to the much faster iSBC 286/100A with 8 MHz speeds up the whole design. Depending on the switch position for the engine subsystems (see Figures 2-3 and 2-4) the total cycle time is in the range of 1380 ms to 1490 ms. Figure 5-1 shows in detail the timing situation of one cycle and how much time was spent in each of the routines. Note the overhead due to the Multibus II transport, and that the message passing handling is not very large. The MPC handling (receive and transmit the message) takes 180 ms, and the swap for the transport adaptation takes 17 ms, which means a total of around 200 ms for the "overhead". Therefore, the bottom line message is very clear: There is enough spare time to add more functions to the engine model. During the timing analysis, 15 tons of "software fuel" was burned!

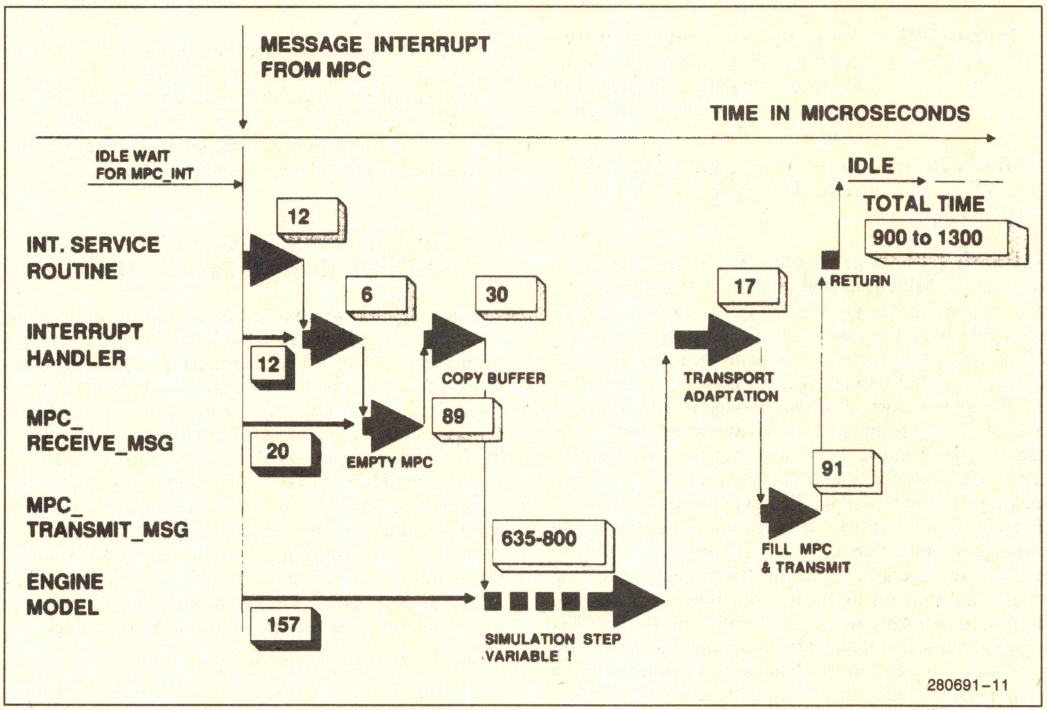


Figure 5-1. The Application Processor Timing Analysis

6.0 SUMMARY AND CONCLUSIONS

The design and implementation of the engine simulation model has given the opportunity to discuss and study a couple of interesting problems around a Multibus II application:

— Given a task, how to partition an application to use more than one processor and assess the problems around it.

— To demonstrate that the port of an existing software written a couple of years ago can still be integrated and ported to the latest hardware technology.

— To adapt an existing Multibus I system to use the benefits of the Multibus II technology.

— To demonstrate the use of the iRMX II operating system together with custom-made software and study its implications when the comm layer is used.

In addition, this application allowed discussion of actual implementation starting from the given problem statement and ending with the final integration in a real system using single board computers. The results show the feasibility, including the important performance analysis.

APPENDIX A

Bibliography and Related Readings

1. Francois Huguenin, "Microcomputer Based Engine Model Used in Flight Simulation Applications", *AIAA Paper 81-0973*, AIAA Flight Simulation Technologies Conference, June 16-18, 1981, Long Beach, CA.
2. F. Huguenin, P. Grepper, "Four-Dimensional Helical Approach of Aircraft in an Air Traffic Control Environment", *AIAA Paper 79-1776R, Journal of Guidance and Control*, June 1981.
3. Furrer + Gloor AG Multibus II MMG
   Furrer + Gloor Silbernstrasse 10
   CH 8953 Dietikon ZH Switzerland
   MBII/LNK-1 Coupler Board User's Guide
4. IEEE 1296 Specification
5. Intel Multibus II Documentation:
   5.1 MPC User's Manual, Order #176526-002
   5.2 Multibus II Transport Specs, Order #453508-001
   5.3 iSBC 286/100A User's Manual, Order #149093-002
   5.4 iRMX II Nucleus Programming Concepts, Order #469151-002
6. iPAT Analyst User's Guide, Order #450583-003

APPENDIX B

Glossary of Terms

Application Terms:

T	Temperature
p	Pressure
m	Mach
HPX	Horse Power Extraction
EN2	Engine Variable N2 Calibrated
EALT	Engine Variable Altitude Calibrated
EWFA	Engine Variable Fuel Flow Calibrated
EVM	Engine Variable Mach Calibrated

Multibus II Terms:

MPC	Message Passing Co-Processor (VLSI Multibus II Interface)
Solicited Message Transfer	A data transfer through Multibus II message space that requires buffer negotiation. May be up to 16 Mbytes long.
Unsolicited Message Transfer	Unsolicited messages arrive at a host unpredictably and can have the effect of an interrupt. Message can have up to 20 bytes of user data.
Dual Port Memory Access	A means of accessing shared memory between two processors. Hardware arbitration is required.
Interconnect Address Access	Access to the Multibus II interconnect address space for test and/or configuration purposes.
Buffer Pool	A collection of memory buffers which are managed by the iRMX II Operating System.
Transport Protocol	This is a generic term describing the function of the software layer that implements the Multibus II Transport Protocol as defined in the IEEE 1296 Specification.

APPENDIX C

Documented Portions of the iRMX II Host Monitor Program

Note that all comments written in bold are added to the original listing portions.

The following code listing contains the most important parts of the iRMX II Host Monitor Program. Since this is a quite large lisiting, it was choosen only to take out which is really important.

B1. Declarations

These word declarations make the interface to the exisiting assembly coded routines.

```
 840   1  =   declare std_conditions word external;
 841   1  =   declare state_1 word external;
 842   1  =   declare state_2 word external;
 843   1  =   declare state_3 word external;
 844   1  =   declare result_1 word external;
 845   1  =   declare result_2 word external;
 846   1  =   declare result_3 word external;
```

The following is an abstract of the definition for the messages which are used between the processors in the system.

```
           =
           =   /*
           =   ; 1. standard conditions message from main to all
                 slaves
           =   */
```

This is the general unsolicited message structure to be used.

```
 847   1  =   declare mpc_message structure
           =   ( dest byte,
           =   src byte,
           =   type byte, not_used byte, message_data (28) byte)
               public;
```

This message will be used for synchronizing all processors together.

```
 848   1  =   declare std_conditions_msg  structure
           =            ( header_std_conditions (4) byte,
           =                transport(4) word,
           =                std_marker word,
           =                evm word,
           =                ealt word,
           =                ep0 word,
           =                et0 word,
           =                dummy_std_conditions (10) word) at
                            (@std_conditions);
```

280691-12

```
These messages are to be used for  the information transfer
between the application processors.

Note that the structures are overlayed to the word defined a lines
840 to 846 !

          =    ; 2.1 Engine #1
          =    */

 849    1  =    declare state_1_msg  structure
          =             ( header_state_1 (4) byte,
          =               transport (4) word,
          =               message_data (10) word) at (@state_1);

          =    ; 3.1 Engine #1
          =    */

 852    1  =    declare result_1_msg  structure
          =             ( header_result_1 (4) byte,
          =               transport (4) word,
          =               message_data (10) word) at (@result_1);
```

280691-13

The following three procedures are used for creating a port, and to send and receive the messages under iRMX II control. See also iRMX II System documentation.

```
1074   1       NEW$PORT: PROCEDURE (port$token$ptr,id,type);
               /*****************************************/
               /*
               this procedure creates a port for access by iRMX II
               */


1075   2          DECLARE port$token$ptr POINTER;
1076   2          DECLARE id word;
1077   2          DECLARE type byte;
1078   2          DECLARE port$token based port$token$ptr token;

                  /*
                  **  Create a new port: port$token$ptr is pointer
                  **  to port$token to be returned
                  **     id is:  for data transport: port ID
                  **             for signal service: message ID
                  **   type is:  port type
                  */

1079   2          if type = data_port then do;
1081   3             port$info.port$id = id;
1082   3             port$info.type = type;
1083   3             port$info.flags = 0;
1084   3             port$token = rq$create$port
                                 (queue$size,@port$info,@status);
1085   3          end;

1086   2          if type = signal_port then do;
1088   3             msg$info.msg$id = id;
1089   3             msg$info.type = type;
1090   3             msg$info.flags = 0;
1091   3             port$token = rq$create$port
                                 (queue$size,@msg$info,@status);
1092   3          end;

1093   2          return;
1094   2       end NEW$PORT;
```

280691-14

```
send a message to the MPC

1095   1       FORWARD$MSG: PROCEDURE (message$pointer,port$id);
1096   2          DECLARE message$pointer POINTER;
1097   2          DECLARE port$id BYTE;

1098   2       DECLARE
                   transaction$id          WORD,
                   status                  WORD;



1099   2          call no$exc$mode;       /*  Set Exception Mode
                                             to 0 (No RMX Action) */

1100   2          transaction$id = rq$send(
                              own$port$tok,sockets(port$id),
                              message$pointer, NIL, 0, 0,
                                          @status);

1101   2          call full$exc$mode;          /*  Reset Exception
                                                   Handling */
1102   2          count$out = count$out+1;     /*  Update Counter for
                                                   messages sent */

1103   2          RETURN;
1104   2       END FORWARD$MSG;
```

280691-15

```
receive a message from the MPC

1105   1      WAIT$FOR$MSG: PROCEDURE (buffer$ptr) WORD;

1106   2         DECLARE buffer$ptr pointer;

1107   2         DECLARE buffer$ptr_origin POINTER;

1108   2         DECLARE
                 info_buf                STRUCTURE(
                                         flags           WORD,
                                         status          WORD,
                                         transaction$id  WORD,
                                         length          DWORD,
                                         forward$port    TOKEN,
                                         socket          DWORD,
                                         message(20)     BYTE,
                                         reserve(4)      BYTE),

                 status                  WORD;


1109   2         call no$exc$mode;    /*  Set Exception Mode to 0
                                          (No RMX Action) */

1110   2         buffer$ptr_origin = rq$receive(
                                       own$port$tok, notimeout,
                                       @info_buf, @status);

1111   2         call full$exc$mode;  /* Reset Exception
                                          Handling*/
                                       /*  Update Counter for
                                          Messages sent */
1112   2         count$in = count$in + 1;
1113   2         socket = info_buf.socket;
                 /*
                 ** We expect just a Control Message ==>
                 ** Copy contents of control message to buffer
                 ** provided by paramter buffer$ptr and return
                 ** sending host ID.
                 */

1114   2         call movb(@info_buf.message,buffer$ptr,20);
1115   2         return socket$def.host$id;

1116   2      END WAIT$FOR$MSG;
```

280691-16

The following code section is the DO UNTIL NO_FUEL loop with the iRMX II System calls used to transfer the data to and receive them back from the Appliaction processors.

```
/*****************************************************************/
/*                                                               */
/*  Wait for state_in Message from LINK                          */
/*  Copy MSG to Buffer =>                                        */
/*                        SWITCH_DECODE,LEVER_TO_2,N2_TRANSIENT  */
/*  Send State_i Message to Engine-Simulators                    */
/*  Wait for Simulators having calculated  ==>                   */
/*           EGT_FILTER, SKAL_BARGRAPHS,SKAL_DIGITS              */
/*  Send display  Message to Link                                */
/*                                                               */

/*****************************************************************/


1366   2          DO WHILE nofuel=0;
```

The state_message contains the data from the I/O subsystem. When received, one simulation step can be processed.

```
                    /* Wait for STATE_IN Message, then Copy */

1367   3            id = WAIT$FOR$MSG(@state_in_msg.state_in);
```

Make the preprocessing.

```
1368   3            CALL SWITCH_DECODE;
1369   3            CALL LEVER_TO_N2;
1370   3            CALL N2_TRANSIENT;
```

Set the marker word to 10H prior to send the data.

```
1371   3            call movb
                         (@(10H),@state_1_msg.message_data,1);
1372   3            call movb
                         (@(10H),@state_2_msg.message_data, 1);
1373   3            call movb
                         (@(10H),@state_3_msg.message_data, 1);

1373   3            call movb (@(10H), @state_3_msg.message_data,
1);
```

280691-17

5

```
1374   3              call forward$msg(
                              @state_1_msg.message_data,engine_1);
1375   3              call forward$msg(
                              @state_2_msg.message_data,engine_2);
1376   3              call forward$msg(
                              @state_3_msg.message_data,engine_3);


                      /*
                      ** Wait for all three having terminated:
                      ** Use logical variables all$done,done$1
                      **   ,done$2,done$3.
                      **
                      ** When a Message arrives, copy it to local
                      ** buffers.
                      */

1377   3              all$done,done$1,done$2,done$3 = FALSE;

1378   3              DO WHILE NOT all$done;

1379   4                 id = WAIT$FOR$MSG(@intermediate);

                         /*
                         ** Message arrived: set corresponnding done
                         ** flag and copy buffer
                         */

1380   4                 if id = host$ids(engine_1) then do;
1382   5                    done$1 = true;
1383   5                    call movb(@intermediate,
                                    @result_1_msg.message_data,28);
1384   5                 end;
1385   4                 if id = host$ids(engine_2) then do;
1387   5                    done$2 = true;
1388   5                    call movb(@intermediate,
                                    @result_2_msg.message_data,28);
1389   5                 end;


1390   4                 if id = host$ids(engine_3) then do;
1392   5                    done$3 = true;
1393   5                    call movb(@intermediate,
                                    @result_3_msg.message_data,28);
1394   5                 end;

1395   4                 all$done = done$1 AND done$2 AND done$3;
1396   4              END;
```

Make the postprocessing.

```
1397   3              CALL EGT_FILTER;
1398   3              CALL SKAL_BARGRAPHS;
1399   3              CALL SKAL_DIGITS;
```

280691-18

```
                /*    send results to MBII/LNK-1 Board with two
                      consecutive unsolicited messages
                */

Set the marker word to 44H and 45H for identification
inside the MBII/LNK-1 Baord.

1400   3              call movb (@(44H),
                         @display_on_line_msg_1.display_on_line,1);
1401   3              call movb (@(45H),
                         @display_on_line_msg_2.display_on_line,1);

Send the two conscutive messages to the MBII/LNK-1 board.

1402   3              CALL FORWARD$MSG(
                            @display_on_line_msg_1.display_on_line,
                                                          link_1);
1403   3              CALL FORWARD$MSG(
                            @display_on_line_msg_2.display_on_line,
                                                          link_1);

                      /*
                      **  Update Message Counter for display on the
                      **  console
                      */

1404   3              status = put$char(@pos$in,fb);
1405   3              status = put$ddec(count$in,fb);
1406   3              status = put$char(@(alloff,0),fb);
1407   3              call put$line(nolf);

1408   3              status = put$char(@pos$out,fb);
1409   3              status = put$ddec(count$out,fb);
1410   3              status = put$char(@(alloff,0),fb);
1411   3              call put$line(nolf);

1412   3              ttime.systime=0;
1413   3              call dq$decode$time(@ttime,@status);
1414   3              if last$sec <> ttime.time(7) then do;
1416   4                 call movb(@pos$time,@outbuff,12);
1417   4                 call movb(@ttime.time,@outbuff(12),8);
1418   4                 call movb(@(alloff),@outbuff(20),4);
1419   4                 out$buff(24)=0;
1420   4                 call disp(@out$buff);
1421   4                 last$sec = ttime.time(7);
1422   4              end;
1423   3           END;
```

280691-19

5

APPENDIX D

Documented Code Listing of the Application Software Based in EPROM

This appendix contains most of the listings generated for the application processor EPROM resident software:

— MAIN05 : Main Module for Cold Start
— MPCDR3 : Low-Level MPC Driver
— C28I6 : Interrupt Handling Routine
— CF650L : Engine Model Module

Note also the last page of the LOCATE MAP which gives an idea on how much memory was needed for this application.

```
iRMX 86 0006/07/08/106 MACRO ASSEMBLER V2.0 ASSEMBLY OF MODULE MAIN_MODULE_286
OBJECT MODULE PLACED IN MAIN05.OBJ
ASSEMBLER INVOKED BY:  :LANG:asm86 MAIN05.A86 DEBUG SYMBOLS TYPE

LOC  OBJ                  LINE     SOURCE

                             1     name    main_module_286
                             2
                             3     ;             PROJECT       CF6-50C Multicomputer
                             4
                             5     ;             08.10.87   v1.0 fh add reset not complete on entry point
                             6     ;             03.10.87   v1.0 fh add in it of ewfaold, efuelused
                             7     ;             21.08.87   v2.1 C8/Scm OCW2 has address pic_cnt10
                             8     ;             04.08.87   v2.0fh  change interrupt structure
                             9     ;             03.08.87   v1.1fh  change external segments
                            10
                            11     ;             03.08.87   v1.0fh  initial start up module
                            12
                            13
                            14     ; This module contains the cold reset start up of the sbc286/100A
                            15     ; and the interrupt processing routine.
                            16
                            17
                            18
                            19     ; note that all written in CAPITAL is original software written
                            20     ; end of 1979, early 1980 '
                            21     ; all other is modification done to port the original application
                            22     ; to Multibus II
                            23
                            24
                            25
                            26
                            27
                            28 +1  $nogen
                            29
                            30     ; interrupt controller PIC
                            31
  00C8                      32     pic_cnt10       equ     0c8h    ; command port 1
  00C2                      33     pic_cnt11       equ     0c2h    ; command port 2
                            34
                            35
----                        36     int_pointers    segment at 0
                            37
0000                        38                     org     0h
0000 (32                    39     type_x          dd      32      dup(?)  ; discard all system interrupts
     ????????
     )
                            40                                             ; which are not used here
                            41
                            42                     ; master pic of iSBC 286/100A
                            43
0000 00000000               44     type_32         dd      0       ; ir 0
0004 00000000               45     type_33         dd      0       ; ir 1
0008 00000000               46     type_34         dd      0       ; ir 2 message interrupt
000C 00000000               47     type_35         dd      0       ; ir 3 iPSB message error
0010 00000000               48     type_36         dd      0       ;
0014 00000000               49     type_37         dd      0       ;
0018 00000000               50     type_38         dd      0       ;
001C 00000000               51     type_39         dd      0       ;
                            52
                            53
----                        54     int_pointers    ends
                            55
                            56     ; external variables needed for initialization
                            57     ; first time after reset
                            58
```

280691-20

```
LOC  OBJ                  LINE     SOURCE

----                        59     EFGIO   SEGMENT COMMON
                            60     extrn   ewfaold:word
                            61     extrn   efuelused:word
                            62     extrn   deltat:word
                            63
----                        64     EFGIO   ENDS
                            65
                            66     $eject
                            67
                            68
----                        69     STACKSEG                SEGMENT STACK
0000 (100                   70                     DW       100 DUP(?)       ;ALLOCATE 100 WORDS FOR STACK
     ????
     )
00C8                        71     STACKTOP        LABEL   WORD             ;OFFSET ADDRESS OF THE TOS
----                        72     STACKSEG                ENDS
                            73
----                        74     iodatseg                SEGMENT public
                            75
----                        76     iodatseg                ENDS
                            77
                            78
                            79     ; two external routines
                            80
                            81     extrn   mpc_init:far            ; initialization of the mpc
                            82     extrn   mpc_interrupt:far       ; mpc interrupt routine
                            83                                     ; for handling I/O and engine
                            84
                            85
----                        86     EFGCOD          SEGMENT PUBLIC
                            87
                            88             ;GIVE ASSEMBLER INITIAL REGISTER TO SEGMENT CORRESPONDENCE
                            89
                            90     ASSUME  CS: EFGCOD,     DS: IODATSEG,   ES: EFGIO,      SS: STACKSEG
                            91
                            92
                            93
0000                        94     interrupt_routine       proc
                            95
                            96
                            97     ; This Interrupt routine will be activated by the MINT line
                            98     ; of the MPC when the MPC has received a message
                            99     ; This routine call the interrupt handler written in PLM86
                           100     ; which will calculate a new state of the engine based on the
                           101     ; incoming message. Afterwards the result message will be formed
                           102     ; and send back to the host.
                           103
0000 50                    104             push    ax
0001 9A0000----      E     105             call    mpc_interrupt
                           106
                           107     ; EOI handling for the 8259A Controller
                           108
0006 B020                  109             mov     al,20h                  ; non specific eoi
0008 E6C0                  110             out     pic_cntl0,al            ; write to ocw2
000A 58                    111             pop     ax
000B FB                    112             sti                             ; enable interrupts
000C CF                    113             iret                            ; return from interrupt
                           114
                           115
                           116     interrupt_routine       endp
                           117
                           118
                           119
000D                       120     start:
                           121
                           122             %reset_interconnect     ; RSTNC# Bit reset inside 8751
                           132                                     ; Microcontroller
                           133
                           134
                           135             %set_int_vector(08h,00h,interrupt_routine)
                           144
                           145     ; set up segment registers of cpu
                           146
0032 B8----          R     147             mov     AX,IODATSEG     ;LOAD SEGMENT REG
0035 8ED8                  148             mov     DS,AX           ;CS IS LOADED AUTOMATICALY BY RESET
0037 B8----          R     149             MOV     AX,EFGIO
003A 8EC0                  150             MOV     ES,AX
003C B8----          R     151             MOV     AX,STACKSEG
003F 8ED0                  152             MOV     SS,AX
0041 BCC800          R     153             MOV     SP,OFFSET STACKTOP  ;SET SP  TO TOS
                           154
                           155             ; cpu now ready for work for the engine model
                           156
                           157             ; but first initialize the mpc
                           158
                           159
0044 9A0000----      E     160             call    mpc_init
                           161
                           162             ; then set to zero fuel variables
                           163
0049 26C706000000  E       164             mov      es:ewfaold,0
0050 26C706000000  E       165             mov      es:efuelused,0
0057 26C70600C5046 E       166             mov      es:deltat,10000
                           167
                           168
                           169             ; then initialize master PIC of iSBC 286/100A
                           170
005E BAC800                171             mov     dx,pic_cntl0    ; icw1
0061 B017                  172             mov     al,00010111b    ; edge triggered, single mode, call
```

280691-21

```
LOC  OBJ                LINE     SOURCE

                        173                                     ; address
0063 EE                 174              out     dx,al
0064 BAC200             175              mov     dx,pic_cntl1   ; icw2
0067 B020               176              mov     al,00100000b   ; base=type 32 (80h)
0069 EE                 177              out     dx,al
006A B01D               178              mov     al,00011101b   ; special fully nested mode, buffered
                        179                                     ; master, normal eoi, 86 mode
006C EE                 180              out     dx,al
006D B0FB               181              mov     al,0fbh        ; enable only ir2 message interrupts
006F EE                 182              out     dx,al
                        183
                        184              ; and finally enable cpu interrupts
                        185
0070 FB                 186              sti
                        187
                        188
0071 90                 189     idle:    nop
0072 90                 190              nop
0073 90                 191              nop                    ; do nothing but wait for message
```

280691-22

```
iRMX 86 PL/M-86 V2.7 COMPILATION OF MODULE C286_INT_HANDLER
OBJECT MODULE PLACED IN C286I6.OBJ
COMPILER INVOKED BY:  :LANG:plm86 C286I6.P86 DEBUG SYMBOLS LARGE TYPE


  1          c286_int_handler:   do;

             /*
             main control interrupt handler for the local engine
             control. Will check for type of message, start up the model
             and carry out one simulation step at a time based on
             incomming messages.
             */

             /*
             history

             c286i6  v1.1    08.10.87    remoce error for std conditions
             c286i6  v1.0    03.10.87    clear mpc_message structure after transmit
                                         and after receiving for next time
             c286i5  v1.0    03.10.87    remove error in adressing mpc_messsage struc
             c286i4  v1.3    03.10.87    normal mode of ops for i2ice test
             c286i4  v1.22   21.08.87    engine and movb removed for test
             c286i4  v1.2    11.08.87    exchange source/dest for retransmit
             c286i4  v1.1    06.08.87    change data strutures
             c286i4  v1.0    06.08.87    change for std_conditions message
             c286i3  v1.1    04.08.87    add source id for the main processor
             c286i3  v1.0    04.08.87    add destination id
             c286i2  v1.0    04.08.87    remove types
             c286i1  v1.0    03.08.87    initial version

             */


  2   1      declare std_conditions word external;
  3   1      declare inputs word external;
  4   1      declare outputs word external;
  5   1      declare byte_dummy   byte;
  6   1      declare word_dummy   word;


  7   1      declare mpc_message structure
             ( dest           byte,              /* 4 Byte Hardware-Header */
               src            byte,
               type           byte,
               not_used       byte,
               prot_id        byte,              /* 8 Byte RMX-Header */
               transm_ctr     byte,
               dest_port_id   word,
               source_port_id word,
               transact_id    byte,
               transact_ctr   byte,
               message_data (20) byte) public; /* 20 Byte User-Data */


  8   1      declare index byte;

             /* Declaration of external procedures ###################*/

  9   1      engine: procedure external;
 10   2      end engine;

 11   1      mpc_receive_message: procedure external;
 12   2      end mpc_receive_message;

 13   1      mpc_transmit_message: procedure external;
 14   2      end mpc_transmit_message;

             /* Start MPC-Interrupt-Procedure (MINT-Signal) ##########*/

 15   1      mpc_interrupt: procedure  public;

             /*

             first empty the MPC FIFO into mpc_message structure

             */

 16   2      call mpc_receive_message;


 17   2      if mpc_message.message_data(0)= 01h

                 then                /* we had a engine init message */
 18   2      do;
 19   3              call movb(@mpc_message.message_data(0),@std_conditions,20);

                    /* clear the mpc_message data structure after receive  */
 20   3               do index = 0 to 19;
 21   4               mpc_message.message_data(index)= 00h;
 22   4               end;
 23   3          end;
```

280691-23

```
24   2          else              /* normal message for one simulation step */

                 if mpc_message.message_data(0) = 10h then
25   2             do;
26   3                call movb (@mpc_message.message_data(0),@inputs,20);

27   3                call engine;      /* one simulation step */

28   3                call movb (@outputs, @mpc_message.message_data(0),20);


29   3                mpc_message.message_data(0)   11h;    /*Set marker for result */

                      /* Transport Protocol adjustment to be
                         conforming to the Comm Layer of iRMX II*/

30   3                byte_dummy        = mpc_message.dest;
31   3                mpc_message.dest = mpc_message.src;
32   3                mpc_message.src  = byte_dummy;

33   3                word_dummy                = mpc_message.dest_port_id;
34   3                mpc_message.dest_port_id  = mpc_message.source_port_id;
35   3                mpc_message.source_port_id = word_dummy;

                      /* transmit results back to host */

36   3                call mpc_transmit_message;

                      /* clear the mpc_message data structure after transmit */
37   3                  do index = 0 to 19;
38   4                  mpc_message.message_data(index)= 00h;
39   4                  end;
40   3                end;



41   2                                                        else
                    do;
                    /* Error-processing: Error in message_data(0)
                       error case not implemented in this version */
42   3                end;



43   2       end mpc_interrupt;


44   1       end;
```

280691-24

```
iRMX 86 8086/87/88/186 MACRO ASSEMBLER V2.0 ASSEMBLY OF MODULE MPC_HANDLER_MODULE
OBJECT MODULE PLACED IN MPCDR3.OBJ
ASSEMBLER INVOKED BY:  :LANG:asm86 MPCDR3.A86 DEBUG SYMBOLS TYPE

LOC  OBJ                  LINE     SOURCE

                             1     name    mpc_handler_module
                             2
                             3     ;       mpcdr3  v1.5 CB/Bcm 21.8.87     reference to mpc_message via 'seg'
                             4     ;       mpcdr3  v1.4 fh 07.08.87        change ds register for far call
                             5     ;       mpcdr3  v1.3 fh 06.08.87        change cl to cx for loop
                             6     ;       mpcdr3  v1.1 fh 06.08.87        rotate message id
                             7     ;       mpcdr3  v1.0 fh 05.08.87        add mctl register
                             8     ;       mpcdr2  v1.1 fh 04.08.87        make procedures far type
                             9     ;       mpcdr2  v1.0 fh 03.08.87        add mpc_init
                            10     ;       mpcdr1  v1.0 fh 22.06.87        add mpc control
                            11     ;       mpcdr0  v1.0 fh 19.06.86        original version
                            12
                            13
                            14     ; This module contains the low level MPC handling routines needed
                            15     ; for this application. Several simplifications are assumed:
                            16     ; the receive and transmit message routines are working on 32 byte
                            17     ; messages only. No length programming assumed. Further, the error
                            18     ; handling is not implemented in this first release of the low level
                            19     ; driver.
                            20     ; all three routines are a direct implementation of the work
                            21     ; flow described in the mpc user's guide and are coded in the large
                            22     ; model for compatibility with the engine model and the plm86
                            23     ; interrupt processing.
                            24     ;
                            25     ;
                            26
                            27
                            28 +1  $include(mpca86.def)
                =1          29     ; i/o port addresses of MPC
                =1          30
                =1          31
  0010          =1          32     mdata   equ     10h     ; message data port
  001C          =1          33     mcmd    equ     1ch     ; message command port
  0000          =1          34     mstat   equ     00h     ; message status register
  0000          =1          35     mrst    equ     00h     ; message reset register
  000C          =1          36     mctl    equ     0ch     ; message control register
  0014          =1          37     merr    equ     14h     ; message error port
  0008          =1          38     mcon    equ     08h     ; message configuration register
                =1          39
  0020          =1          40     msocmp  equ     20h     ; sol message completion port
  0024          =1          41     msicmp  equ     24h     ; sol message input completion port
  0020          =1          42     msocan  equ     20h     ; sol message output cancel port
  0024          =1          43     msican  equ     24h     ; sol message input cancel port
                =1          44
  0004          =1          45     mid     equ     04h     ; message id register
                =1          46
  0030          =1          47     icadr1  equ     30h     ; interconnect address low
  0034          =1          48     icadrh1 equ     34h     ; interconnect address high1
  0031          =1          49     icadrh2 equ     31h     ; interconnect address high2
  003C          =1          50     icdata  equ     3ch     ; interconnect data register
                =1          51
                =1          52     ; bit definitions
                =1          53
  0001          =1          54     xmtnf   equ     1               ; transmit FIFO not full
  0004          =1          55     xmterr  equ     4               ; transmit error on PSB
  0003          =1          56     rcvne   equ     10000011b       ; receive FIFO not empty
                =1          57
  0080          =1          58     init_done       equ     80h     ; filter for bit 7
  0000          =1          59     zero            equ     00h
  0085          =1          60     width_config    equ     10000101b       ; set 16 bit fifo and dma
                =1          61                                             ; full message passing
                =1          62
  0002          =1          63     mpc_control     equ     00000010b       ; receive interrupt enable
                            64
                            65     extrn   mpc_message:word
                            66
                            67
----                        68     code    segment
                            69     assume  cs:code
                            70     assume  ds:seg mpc_message
                            71
                            72     public  mpc_receive_message,mpc_transmit_message,mpc_init
                            73
                            74
0000                        75     mpc_transmit_message    proc far
                            76
                            77     ; v1.1 mpc_write_message                add transmit FIFO control
                            78     ; v1.0 mpc_write_message                basic priciple written
                            79
                            80     ; mpc_transmit message will copy the data from the mpc_message
                            81     ; structure to the fifo of the mpc and send it out to the iPSB.
                            82     ; 32 byte length assumed.
                            83
0000 1E                     84             push    ds
0001 B8----          E      85             mov     ax,seg mpc_message
0004 8ED8                   86             mov     ds,ax
                            87
0006 B91000                 88             mov     cx,16           ; 32 byte message
                            89                                     ; 16 words
0009 BA1000                 90             mov     dx,mdata        ; data port of mpc
000C BE0000          E      91             mov     si, offset mpc_message
                            92
```

280691-25

```
LOC  OBJ                  LINE     SOURCE

000F 8B04                   93     write_1: mov     ax, [si]        ; load data to ax
0011 EF                     94              out     dx,ax           ; store to mpc
0012 46                     95              inc     si
0013 46                     96              inc     si              ; double to next word
0014 E2F9                   97              loop    write_1         ; continue till all 32 byte
                            98                                      ; writte to MPC FIFO
                            99
0016 32C0                  100              xor     al,al           ;
0018 E61C                  101              out     mcmd,al         ; write 00 to MPC CMD Register
                           102
001A                       103     continue:
                           104
001A E400                  105              in      al,mstat        ; read MPC status_register
001C 2401                  106              and     al,xmtnf        ; mask transmit fifo bit
001E 3C01                  107              cmp     al,1
0020 7409                  108              je      transmit_complete       ; message transmitted '
0022 2404                  109              and     al,xmterr       ; mask transmit error bit
0024 3C00                  110              cmp     al,0
0026 74F2                  111              jz      continue        ; message transmition still in progress
                           112
0028 E85600                113              call    transmit_error  ; error handling
                           114
002B                       115     transmit_complete:
                           116
002B 1F                    117              pop     ds
002C CB                    118              ret
                           119
                           120     mpc_transmit_message    endp
                           121
                           122 +1  $eject
                           123
                           124
                           125
002D                       126     mpc_receive_message     proc far
                           127
                           128     ; v1.1 mpc_read_message                 add receive FIFO control
                           129     ; v1.0 mpc_read_message                 basic priciple written
                           130
                           131     ; mpc_receive_message will copy the data to the mpc_message
                           132     ; structure from the fifo of the mpc.
                           133     ; 32 byte length assumed.
                           134
002D 1E                    135              push    ds
002E B8----          E     136              mov     ax,seg mpc_message
0031 8ED8                  137              mov     ds,ax
                           138
0033                       139     not_ready_1:
                           140
0033 E400                  141              in      al,mstat        ; 8 bit read of MPC_status_register
0035 3C83                  142              cmp     al,rcvne
0037 75FA                  143              jne     not_ready_1     ; wait until flag set
                           144
0039 E410                  145              in      al,mdata        ; read one byte for length
                           146
                           147     ; read mpc_fifo  assuming 32 byte valid data and 16 bit transfers
                           148     ; initialized inside the MPC
                           149
003B B91E00                150              mov     cx,16           ; 32 byte message
                           151                                      ; 16 words
003E BA1000                152              mov     dx,mdata        ; data port of mpc
0041 BF0000          E     153              mov     di, offset mpc_message
0044 ED                    154     read_2:  in      ax,dx           ; read from mpc
0045 8905                  155              mov     [di],ax         ; store data to message buffer
0047 47                    156              inc     di
0048 47                    157              inc     di              ; double to next word
0049 E2F9                  158              loop    read_2          ; continue till all 32 byte
                           159                                      ; read
                           160
004B E41C                  161              in      al,mcmd         ; read MPC_command port for resync
                           162
                           163                                      ; service of message read now complete
                           164
                           165
004D 1F                    166              pop     ds
                           167                                      ; continue with processing of data
004E CB                    168              ret
                           169
                           170     mpc_receive_message     endp
                           171
                           172
                           173 +1  $eject
                           174
                           175
004F                       176     mpc_init        proc far
                           177
                           178
                           179     ; this routine will initialize the MPC for use in this application.
                           180
004F 1E                    181              push    ds
0050 B8----          E     182              mov     ax,seg mpc_message
0053 8ED8                  183              mov     ds,ax
                           184
```

280691-26

```
LOC  OBJ                  LINE    SOURCE

                           185
0055 B000                  186             mov     al,zero
0057 E600                  187             out     mrst,al         ; reset the mpc
                           188
0059                       189     chck_reset:
0059 E400                  190             in      al,mstat
005B 2480                  191             and     al,init_done
005D 3C00                  192             cmp     al,zero
005F 75F8                  193             jne     chck_reset      ; do whil not reset
                           194
0061 B085                  195             mov     al,width_config
0063 E600                  196             out     mcon,al         ; set to 16 bit
0065 B002                  197             mov     al,mpc_control  ; interrupt control
0067 E60C                  198             out     mctl,al
0069 E408                  199             in      al,mcon         ; no check for error case this time
006B B0F8                  200             mov     al,0f8h
006D E634                  201             out     icadrh1,al      ; interconnect address high
006F E631                  202             out     icadrh2,al      ; interconnect address high
0071 B098                  203             mov     al,098h
0073 E630                  204             out     icadrl ,al      ; interconnect address low
0075 E43C                  205             in      al,3ch          ; read interconnect register slot_id
0077 D0E8                  206             shr     al,1
0079 D0E8                  207             shr     al,1
007B D0E8                  208             shr     al,1
007D E604                  209             out     mid,al          ; set the message id to slot_id
                           210
007F 1F                    211             pop     ds
0080 CB                    212             ret
                           213
                           214
                           215
                           216     mpc_init        endp
                           217
                           218
0081                       219     transmit_error  proc
                           220
                           221     ; this routine is  n o t  implemented in this release of
                           222     ; this handler
                           223     transmit_error  endp
                           224
                           225
----                       226     code    ends
                           227
                           228
```

280691-28

```
iRMX 86 8086/87/88/186 MACRO ASSEMBLER V2.0 ASSEMBLY OF MODULE CF6_ENGINE_LOCAL
OBJECT MODULE PLACED IN CF650L.OBJ
ASSEMBLER INVOKED BY:  :LANG:asm86 CF650L.A86 DEBUG SYMBOLS TYPE

LOC  OBJ                  LINE     SOURCE

                             1     NAME CF6_engine_local
                             2
                             3
                             4     ;                    ########################################
                             5     ;                    ########################################
                             6     ;                    ##                                    ##
                             7     ;                    ##       ENGINE   MAIN-PROGRAM        ##
                             8     ;                    ##                                    ##
                             9     ;                    ########################################
                            10     ;                    ########################################
                            11
                            12     ;              PROJECT       CF6-50C Multicomputer
                            13
                            14     ;              03.10.87   v1.0    put n2lim as a constant inside routine
                            15     ;              11.08.87   v2.2cg  correct flags
                            16     ;              04.08.87   v2.1fh  make engine procedure far
                            17     ;              03.08.87   v2.0fh  remove local start
                            18     ;              21.07.87   v1.2fh  add for main module test version
                            19     ;              19.06.87   v1.1fh  adapt to local processor architecture
                            20     ;              19.06.87   v1.0fh  add message structure
                            21
                            22
                            23     ;       PROGRAM FOR THE SIMULATION OF THE CF6-50C ENGINE. FOLLOWING
                            24     ;       SUBSYTEM MODULES ARE USED:
                            25     ;                            EPR-SYSTEM
                            26     ;                            EGT-SYSTEM
                            27     ;                            FUEL-FLOW-SYSTEM
                            28     ;                            THRUST-SYSTEM
                            29     ;                            N1-SYSTEM
                            30     ;                            INLET SYSTEM
                            31
                            32     ;       SEE ALSO ADDITIONAL DOCUMENTATION IN THE WRITTEN DOCUMENTS
                            33     ;       SIMULATION OF A JET ENGINE USING A MICROCOMPUTER
                            34     ;       DIPLOMA WORK          WS 1979/80
                            35
                            36     ;*************************************************************************
                            37     ;
                            38     ;     THIS FROGAM USES FOLLOWING EXTERNAL MODULE: CF6LIB.LIB(SYSTEM-MODULE),
                            39     ;       1.CF6LIB.LIB (SYSTEM-MODULE)
                            40     ;       2.TABLIB.LIB (DATA FUNCTIONS-TABLES )
                            41     ;*************************************************************************
                            42
                            43
                            44
                            45     ; note that all written in CAPITAL is original software written
                            46     ; end of 1979, early 1980 '
                            47     ; all other is modification done to port the original application
                            48     ; to Multibus II
                            49
                            50     public std_conditions
                            51
                            52     PUBLIC EVM,EVML,EVMH,EALT,EALTL,EALTH,EP0,ET0,ENGN,INPUTS
                            53     PUBLIC ET2,ED2,ETHE,EDAM,ERTHE,DEPR,DEGT,DEWFA,DN1,DFNA
                            54
                            55     PUBLIC EN2,EN2L,EN2H,OUTPUTS,EN1,ENIL,EN1H,EN1A
                            56     PUBLIC EPR,EGT,EWFA,EFUELUSED,EFNA
                            57     PUBLIC EACPSW,FHPXSW,CFAISW,ECAISW,EBLESW,EREVSW
                            58     PUBLIC EWFAOLD,DELFAT
                            59
                            60     ;             ##############################
                            61     ;             #                            #
                            62     ;             #  DATA VARIABLES            #
                            63     ;             #                            #
                            64     ;             ##############################
                            65
----                        66     EFGIO   SEGMENT COMMON
                            67
                            68     ;           ########################################
                            69     ;           #  WORK DATA OF ENGINE MODEL           #
                            70     ;           ########################################
                            71
                            72     ; global engine input variables
                            73
                            74     ; NOTE THE USE OF ACCESS BOTH AS BYTE AND WORD
                            75
  0000                      76     std_conditions  equ     this word
0000 (1                     77                     dw      1       dup(0)                  ; Flag-Word only used to syn
     0000                          hronize
     )
                            78                                                             ; sim step from initi
                                   alization
                            79
  0002                      80     EVM     EQU     THIS WORD           ;MACH               (INPUT)
0002 00                     81     EVML    DB      0                   ;                     "
0003 00                     82     EVMH    DB      0                   ;                     "
  0004                      83     EALT    EQU     THIS WORD           ;ALTITUDE           "
0004 00                     84     EALTL   DB      0                   ;                     "
0005 00                     85     EALTH   DB      0                   ;                     "
0006 0000                   86     EP0     DW      0                   ;AMBIENT PRESSURE     "
0008 0000                   87     ET0     DW      0                   ;AMBIENT  TEMPERATURE "
000A (10                    88     reserved        DW  10 dup(0)           ;10 byte reserved (multibus II
     0000
     )
```

280691-29

```
LOC  OBJ                  LINE     SOURCE

                            89                                                    ; data strucutre u
                                     f the message
                            90
                            91
                            92     ; local engine variables
                            93
001E 0000                   94     ET2             DW      0
0020 0000                   95     ED2             DW      0
0022 0000                   96     ETHE    DW      0
0024 0000                   97     EDAM    DW      0
0026 0000                   98     ERTHE   DW      0
                            99             ;
                           100     ;TEMP VARIABLES
0028 0000                  101     DEPR    DW      0
002A 0000                  102     DEGT    DW      0
002C 0000                  103     DEWFA   DW      0
002E 0000                  104     DNI             DW      0
0030 0000                  105     DFNA    DW      0
                           106             ;
                           107             ;N2 LOW LIMIT CONSTANT = 80
                           108
0032 FF0F                  109     EN2LIM  DW      4095    ;SCALED VALUE FOR LOWER LIMIT N2
                           110     ;
                           111     ;
                           112
                           113
                           114 +1  $eject
                           115
                           116     ; message data buffers for MPC communication
                           117
  0034                     118     INPUTS EQU      THIS WORD       ;                       "
0034 (1                    119                     dw              1 dup(0)        ; Flag-Word only
     0000
     )
                           120
  0036                     121     EN2     EQU     THIS WORD       ;REFERRED CORE SPEED    "
0036 00                    122     EN2L    DB      0               ;                       "
0037 00                    123     EN2H    DB      0               ;                       "
0038 0000                  124     EACPSW  DW      0               ;AIR COND. SWITCH       "
003A 0000                  125     EHPXSW  DW      0               ;HORSE POWER EX. SWITCH "
003C 0000                  126     EFAISW  DW      0               ;FULL ANTI ICE SWITCH   "
003E 0000                  127     ECAISW  DW      0               ;COWL ANTI ICE SWITCH   "
0040 0000                  128     EBLESW  DW      0               ;BLEED SWITCH           "
0042 0000                  129     EREVSW  DW      0               ;REVERSER SWITCH        "
0044 (2                    130     in_reserved     dw      2 dup(0)        ; reserved for future use
     0000
     )
                           131
                           132
                           133
                           134
0048 0000                  135     ENGN    DW      0               ;TRIEBWERKNUMMER        "
                           136
                           137
                           138
                           139
  004A                     140     OUTPUTS EQU     THIS WORD
004A (1                    141                     dw              1 dup(0)        ; Flag-Word only
     0000
     )
                           142
004C 0000                  143     EWFAOLD DW      0               ;"OLD VALUE FUEL FLOW   OUTPUT
004E 0000                  144     DELTAT  DW      0               ; SAMPLING TIME         "
0050 0000                  145     EFUELUSED       DW      0       ; FUEL USED             "
  0052                     146     EN1     EQU     THIS WORD       ; REFERD FAN SPEEDD     "
0052 00                    147     EN1L    DB      0
0053 00                    148     EN1H    DB      0
0054 0000                  149     EN1A    DW      0               ; N1 DIGITAL DSIPLAY    "
0056 0000                  150     EPR     DW      0               ; ENG.PRESSURE RATIO     "
0058 0000                  151     EGT     DW      0               ; EXHAUST GAS TEMP.      "
005A 0000                  152     EWFA    DW      0               ; FUEL-FLOW              "
005C 0000                  153     EFNA    DW      0               ; THRUST                 "
                           154
                           155
005E 0000                  156     EFUELUSED1      dw      0
0060 0000                  157     EFUELUSFD2      dw      0
0062 0000                  158     EFUELUSED3      dw      0
                           159
----                       160     EFGIO   ENDS
                           161
                           162 +1  $eject
                           163
                           164
----                       165     STACKSEG                SEGMENT STACK
0000 (100                  166                     DW      100 DUP(?)      ;ALLOCATE 100 WORDS FOR STACK
     ????
     )
00C8                       167     STACKTOP        LABEL   WORD            ;OFFSET ADDRESS OF THE TOS
----                       168     STACKSEG                ENDS
                           169
----                       170     iodatseg                SEGMENT public
                           171
0000 0000                  172     TIMES0  dw      0
0002 00                    173     NOFUEL  db      0
                           174
----                       175     iodatseg                ENDS
                           176
                           177
```

280691-30

```
LOC  OBJ                  LINE    SOURCE

----                       178    EFGCOD          SEGMENT PUBLIC
                           179
                           180            ;GIVE ASSEMBLER INITIAL REGISTER TO SEGMENT CORRESPONDENCE
                           181
                           182    ASSUME  CS: EFGCOD,     DS: IODATSEG,   ES: EFGIO,      SS: STACKSEG
                           183
                           184    public  engine
                           185
                           186
                           187
                           188
                           189
                           190    EXTRN INLETSYS:NEAR,EPRSYS:NEAR,EGTSYS:NEAR,N1SYS:NEAR,EWFASYS:NEAR
                           191    EXTRN FNASYS:NEAR
                           192
                           193            ASSUME ES:EFGIO,CS:EFGCOD
                           194
                           195    ;                       ##############################
                           196    ;                       #                            #
                           197    ;                       # ENGINE MODEL ROUTINE       #
                           198    ;                       #                            #
                           199    ;                       ##############################
                           200
                           201
0000                       202    ENGINE          PROC    far
                           203
0000                       204    START_engine:
0000 06                    205                            PUSH    ES
0001 1E                    206                    PUSH    DS          ;ES,DS SAVE ONTO STACK
0002 B8----          R     207                    MOV     AX,EFGIO    ; REORGANIZE SEGEMENTS FOR THE
                           208                                                    ; OPTIMAL USE OF THE CPU :
                               SPEED AND
                           209                                                    ; AGAIN SPEED....
0005 8EC0                  210                    MOV     ES,AX
0007 E80000          E     211                    CALL    INLETSYS    ;CALCULATE THE INLET VALUES
                           212
                           213
                           214
                           215
                           216    ;               ##################
                           217    ;               #    ENGINE      #
                           218    ;               ##################
                           219
                           220
000A 26A13600              221                    MOV     AX,EN2
000E 3DFF0F                222                    CMP     AX,4095         ; >>> put absolute value here
                           223                                                    ; due to EPROM version
0011 7706                  224                    JA      NORMAL2
0013 E81500                225                    CALL    IDLEBYPASS      ; IN IDLE BYPASS MODEL CALCULATION
                           226                                                    ; BASED ON A SIMPLIFICATION A
                               SSUPTION
0016 E81090                227                    JMP     idle_state
                           228
                           229                    ; NORMAL NON IDLE MODE : CALCULATE A NEW STATE OF THE ENGINE
                           230                    ; SUBSYSTEMS
                           231
0019 E80000          E     232    NORMAL2:        CALL    EPRSYS          ; EPR
001C E80000          E     233                    CALL    N1SYS           ; N1
001F E80000          E     234                    CALL    EWFASYS         ; FUEL FLOW
0022 E80000          E     235                    CALL    EGTSYS          ; EGT
0025 E80000          E     236                    CALL    FNASYS          ; THRUST
                           237
0028                       238    idle_state :
                           239
0028 1F                    240                    POP     DS
0029 07                    241                    POP     ES
002A CB                    242                    RET
                           243    ENGINE  ENDP
                           244
                           245    ;                       ################
                           246    ;                       #  END ENGINE  #
                           247    ;                                       ################
                           248
002B                       249    IDLEBYPASS      PROC
                           250
002B 50                    251            PUSH    AX              ;SAVE EN2
002C 26C7063600000010      252            MOV     EN2,4096
                           253
0033 E80000          E     254            CALL    EPRSYS          ;N1SYS SIMPLIFICATION ASSUMPTION
0036 E80000          E     255            CALL    EWFASYS
0039 E80000          E     256            CALL    EGTSYS
003C E80000          E     257            CALL    FNASYS
                           258
003F 268F063600            259            POP     EN2             ;RECALL EN2
0044 C3                    260            RET
                           261
                           262    IDLEBYPASS      ENDP
                           263
                           264
                           265
----                       266    EFGCOD  ENDS
                           267            END
```

280691-31

```
MODULE = FUELFLOW_SYSTEM

BASE    OFFSET TYPE SYMBOL              BASE    OFFSET TYPE SYMBOL              BASE    OFFSET TYPE SYMBOL

0041H   0A10H  SYM  CONT1               0041H   0A33H  SYM  CONT2               0041H   0C0BH  SYM  ENDE
0041H   0B25H  SYM  EWFAC              0041H   09C0H  SYM  EWFASYS             0041H   0A06H  SYM  NEG1
0041H   0A29H  SYM  NEG2              0041H   09C0H  SYM  START              0041H   0AABH  SYM  TACP
0041H   0ACEH  SYM  TCAI              0041H   0A6CH  SYM  TFAI               0041H   0A8DH  SYM  THPX


MEMORY MAP OF MODULE MAIN_MODULE_286

MODULE START ADDRESS  PARAGRAPH = 0041H  OFFSET = 000DH
SEGMENT MAP

START     STOP      LENGTH ALIGN NAME            CLASS           OVERLAY

00000H    0009FH     00A0H   A   INT_POINTERS
00200H    00200H     0000H   G   ??SEG
00200H    00263H     0064H   G   EFGIO
00270H    003FFH     0190H   G   STACKSEG
00400H    00402H     0003H   G   IODATSEG
00410H    01021H     0C12H   G   EFGCOD
01030H    010B0H     0081H   G   CODE
010C0H    0243FH     1380H   G   EFGDAT
02440H    0251EH     00DFH   W   C286_INT_HANDL  CODE
                               -ER_CODE
02520H    02543H     0024H   W   C286_INT_HANDL  DATA
                               -ER_DATA
02544H    0254BH     0008H   W   STACK           STACK
0254CH    0254CH     0000H   W   MEMORY          MEMORY
```

280691-32

5

APPENDIX E

I2ICE Debug Session Sample

This shows a sample debug session using symbolic debugging with I2ICE 286.

```
First include all user pre-defined macros for this application
*
*
*include cf6.mac

std_def is a macro for setting the standard values for local
engine test

*define proc std_def = do
.*evm=0t
.*ep0=20783t
.*et0=303t
.*deltat=18000t
.*end
*
mpc_msg is a macro for displaying the content of the MPC MESSAGE
STRUCTURE

*define  proc mpc_msg = do
.*write ' '
.*write ' dest      = ', mpc_message.dest
.*write ' src       = ', mpc_message.src
.*write ' type      = ', mpc_message.type
.*write ' not_used = ', mpc_message.not_used
.*write ' transport message passing part '
.*write ' protocol_id       = ', byte .mpc_message.prot_id
.*write ' transmition _cntl = ', byte .mpc_message.transm_ctr
.*write ' dest port_id      = ', word .mpc_message.dest_port_id
.*write ' src_port_id       = ', word .mpc_message.sourc_port_id
.*write ' transaction_id    = ', byte .mpc_message.transact_id
.*write ' transaction_cntl  = ', byte .mpc_message.transact_ctr
.*write ' data field = '
.*word .mpc_message+12t length 10t
.*end

macro s makes a single step

*define proc s = do
.*istep
.*ASM $
.*end
*

macro input_cf6 displays the inputs data structure
before and engine simulation step
```

280691-33

```
*define proc input_cf6 = do
.*write 'flag_word =',:cf6_engine_local.inputs
.*write 'EN2 =',:cf6_engine_local.EN2
.*write 'EN2L =',:cf6_engine_local.EN2L
.*write 'EN2H =',:cf6_engine_local.EN2H
.*write 'EACPSW =',:cf6_engine_local.EACPSW
.*write 'EHPXSW =',:cf6_engine_local.EHPXSW
.*write 'EFAISW =',:cf6_engine_local.EFAISW
.*write 'ECAISW =',:cf6_engine_local.ECAISW
.*write 'EBLESW =',:cf6_engine_local.EBLESW
.*write 'EREVSW =',:cf6_engine_local.EREVSW
.*end

*

macro output_cf6 displays the outputs data structure
as a result of a simulation step

*
*define proc output_cf6 = do
.*write 'flag_word =',:cf6_engine_local.outputs
.*write 'EWFAOLD =',:cf6_engine_local.EWFAOLD
.*write 'DELTAT =',:cf6_engine_local.DELTAT
.*write 'EFUELUSED =',:cf6_engine_local.EFUELUSED
.*write 'EN1  =',:cf6_engine_local.EN1
.*write 'EN1L  =',:cf6_engine_local.EN1L
.*write 'EN1H  =',:cf6_engine_local.EN1H
.*write 'EN1A  =',:cf6_engine_local.EN1A
.*write 'EPR =',:cf6_engine_local.EPR
.*write 'EGT =',:cf6_engine_local.EGT
.*write 'EWFA =',:cf6_engine_local.EWFA
.*write 'EFNA =',:cf6_engine_local.EFNA
.*end
*

enginestate decode the full state of variables of the
engine model at any time

*DEFINE proc ENGINESTATE = do
.*BASE=decimal
.*WRITE 'N1 =  ',EN1, '  non scaled :',(en1a/1696t)*20t
.*WRITE 'EGT = ',EGT, '  non scaled :',egt
.*WRITE 'EPR = ',EPR, '  non scaled :',epr/3200t
.*WRITE 'N2 =  ',EN2, '  non scaled :',(en2/256t)*5t
.*WRITE 'FF =  ',EWFA,'  non scaled :',ewfa
.*WRITE 'FNA = ',EFNA,'  non scaled :',(efna/378t)*1000t
.*WRITE 'INLET ::::   EP0=',EP0,' ET0= ',ET0,' MACH = ',EVM
.*write ' '
.*write ' switch status '
.*write 'acp =',EACPSW,' hpx =',EHPXSW,' fai =',EFAISW
.*write 'cai =',ECAISW,' ble =',EBLESW,' reverse = ',EREVSW
.*write ' fuel used for this step = ',efuelused
.*BASE=Hex
.*end
```

280691-34

```
macro setn2 sets the en2 variable for a local test
*
*DEFINE proc SETN2 = do
.*word .mpc_message+14t = %0
.*end
*

macro rst allows the reset not complete sequence to take place
see also main text

*define proc rst = do
.*reset regs
.*unithold
.*port(34h)=0f8h
.*port(31h)=0f8h
.*port(30h)=0a4h
.*port(3ch)=000h
.*go from 0ffff:0 forever
.*end
*


make a break at the end of the simulation step

*go til :c286_int_handler#43
*Probe 0 stopped at :C286_INT_HANDLER#41 + 1H because of execute
break
  Clips= F3  Trace Buffer Overflow

see what data are around after a full engine simulation step
using i2ice macros

incoming data packed in mpc_message structure as before
engine simulation step done

message strucuture as received by the MPC

*mpc_msg

 dest      =  01
 src       =  05
 type      =  00
 not_used  =  00
 transport message passing part
 protocol_id        =  02
 transmition _cntl  =  00
 dest port_id       =  0010
 src_port_id        =  0800
 transaction_id     =  00
 transaction_cntl   =  00
 data field =
0256:000EH  0010 14E0 0000 0001 0001 0000 0000 0000 0000 0000
```

280691-35

```
input data

*input_cf6
flag_word = 0010
EN2 = 14E0
EN2L = E0
EN2H = 14
EACPSW = 0000
EHPXSW = 0001
EFAISW = 0001
ECAISW = 0000
EBLESW = 0000
EREVSW = 0000

output data

*
*output_cf6
flag_word = 0000
EWFAOLD = 1411
DELTAT = 29C7
EFUELUSED = 001B
EN1  = 2CDB
EN1L  = DB
EN1H  = 2C
EN1A  = 2605
EPR = 5346
EGT = 035B
EWFA = 5047
EFNA = 4B0A

full engine state

*
*enginestate
N1 =   11483   non scaled : 100
EGT =  859   non scaled : 859
EPR =  21318   non scaled : 6
N2 =   5344   non scaled : 100
FF =   20551   non scaled : 20551
FNA =  19210   non scaled : 50000
INLET ::::   EP0= 20783  ET0=  303  MACH =  0

 switch status
acp = 0  hpx = 1  fai = 1
cai = 0  ble = 0  reverse =  0
 fuel used for this step =  27
*
```

280691-36

5

```
continue the simulation
*
*go forever
?
?
?
/* end of this debug session */
```

280691-37

iPAT Analysis Protocol

This shows the sample iPAT analysis session done to assess the performance of the application processor design.

```
*
*/* -------ipat analysis    --------*/
*
set the time base to 200 nsec in general
note that we are online with the simulation running

*go
?ptimebase = 200ns

define a macro for analysing

?
?define proc measure = do
.?pat init duration %0
.?histo=false
.?pat display
.?end
?

measure the time spendt in the mpc_receive_message routine

?measure (:mpc_handler_module.mpc_receive_message)

Mode: DURATION
Event: :MPC_HA.MPC_RECEIVE_MESSAGE
Time Range: 200ns TO 1sec
PTIMEBASE = 200 ns
Status: OK

    Time Interval :  Bin Sum : Time    ]
------------------+----------+--------]
        <  200 ns :         0:     0 ns]
  200 ns- 1200 ns+:         0:     0 ns]
 1400 ns- 9200 ns+:         0:     0 ns]
 9400 ns-   65 us+:         0:     0 ns]
   65 us-  447 us+:        58:  5165 us]
------------------+----------+--------]
Time Min:    89.0 us
Time Max:    89.2 us
?
```

280691-38

```
measure the time spent in the mpc_transmit_message routine


?measure (:mpc_handler_module.mpc_transmit_message)

Mode: DURATION
Event: :MPC_HA.MPC_TRANSMIT_MESSAGE
Time Range: 200ns TO 1sec
PTIMEBASE = 200 ns
Status: OK

    Time Interval :  Bin Sum : Time    ]
------------------+----------+--------]
        <  200 ns :         0:     0 ns]
  200 ns- 1200 ns+:         0:     0 ns]
 1400 ns- 9200 ns+:         0:     0 ns]
 9400 ns-   65 us+:         0:     0 ns]
   65 us-  447 us+:        58: 5316 us]
------------------+----------+--------]
Time Min:   91.6 us
Time Max:   91.8 us
?


measure the time spent in the whole interrupt  routine

?
?measure (:c286_int_handler)

Mode: DURATION
Event: :C286_INT_HANDLER
Time Range: 200ns TO 1sec
PTIMEBASE = 200 ns
Status: OK

    Time Interval :  Bin Sum : Time    ]
------------------+----------+--------]
        <  200 ns :         0:     0 ns]
  200 ns- 1200 ns+:         0:     0 ns]
 1400 ns- 9200 ns+:         0:     0 ns]
 9400 ns-   65 us+:         0:     0 ns]
   65 us-  447 us+:         0:     0 ns]
------------------+----------+--------]
Time Min: 1409.2 us
Time Max: 1412.8 us
?

measure the time spent in the engine model routine
note that the engine model calculation time is variable depending
on the switch position of the subsystems

?measure (:cf6_engine_local.engine)
```

280691-39

```
Mode: DURATION
Event: :CF6_EN.ENGINE
Time Range: 200ns TO 1sec
PTIMEBASE = 200 ns
Status: OK

    Time Interval :  Bin Sum : Time    ]
------------------+----------+--------]
        <  200 ns :         0:     0 ns]
  200 ns- 1200 ns+:         0:     0 ns]
 1400 ns- 9200 ns+:         0:     0 ns]
 9400 ns-   65 us+:         0:     0 ns]
   65 us-  447 us+:         0:     0 ns]
------------------+----------+--------]
Time Min:  633.0 us
Time Max:  636.6 us
?

measure the time spent for copying the data from the
mpc_message_structure to the inputs buffer


?
?pat init duration :c286_int_handler#26 to :c286_int_handler#27
?pat display

Mode: DURATION
Event: :C286_INT_HAN#26-#27
Time Range: 200ns TO 1sec
PTIMEBASE = 200 ns
Status: OK

    Time Interval :  Bin Sum : Time    ]
------------------+----------+--------]
        <  200 ns :         0:     0 ns]
  200 ns- 1200 ns+:         0:     0 ns]
 1400 ns- 9200 ns+:         0:     0 ns]
 9400 ns-   65 us+:        62: 1895 us]
   65 us-  447 us+:         0:     0 ns]
------------------+----------+--------]
Time Min:   30.4 us
Time Max:   30.6 us

?

measure the time spent for the transport protocol adaptation

?pat init duration :c286_int_handler#30 to :c286_int_handler#35
?pat display

Mode: DURATION
Event: :C286_INT_HAN#30-#35
Time Range: 200ns TO 1sec
PTIMEBASE = 200 ns
Status: OK
```

280691-40

5

```
    Time Interval :  Bin Sum : Time    ]
------------------+----------+--------]
        <  200 ns :         0:     0 ns]
  200 ns- 1200 ns+:         0:     0 ns]
 1400 ns- 9200 ns+:         0:     0 ns]
 9400 ns-   65 us+:        50:   873 us]
   65 us-  447 us+:         0:     0 ns]
------------------+----------+--------]
Time Min:    17.4 us
Time Max:    17.6 us
?
```

measure the total time used for the mpc interrupt processing from hardware interrupt on.

```
Mode: DURATION
Event: :MAIN_M.INTERRUPT_ROUTINE
Time Range: 200ns TO 1sec
PTIMEBASE = 200 ns
Status: OK

    Time Interval :  Bin Sum : Time    ]
------------------+----------+--------]
        <  200 ns :         0:     0 ns]
  200 ns- 1200 ns+:         0:     0 ns]
 1400 ns- 9200 ns+:         0:     0 ns]
 9400 ns-   65 us+:         0:     0 ns]
   65 us-  447 us+:         0:     0 ns]
------------------+----------+--------]
Time Min: 1450.0 us
Time Max: 1453.6 us
```

profile the relative time spend by the three main routines

```
?
?pat init profile :mpc_handler_module.mpc_receive_message,&
??:mpc_handler_module.mpc_transmit_message,&
??:cf6_engine_local.engine
?pat display

Mode: PROFILE
PTIMEBASE =  10 us
Include calls
Status: OK
```

280691-41

```
Event              :   Count  0%        10         20
---------------------------+---------+---------+-------
ENGINE             :         61:xxxxxxxxxxxxxxxx
MPC_TRANSMIT_ME    :         61:xxxxxxxxxxxxxxxx
MPC_RECEIVE_MES    :         61:xxxxxxxxxxxxxxxx
*Background*       :        182:xxxxxxxxxxxxxxxxxxxxxxxxxxxx
---------------------------+---------+---------+-------
      Total        :        365 0%        10         20
?histo=false
?pat display

Mode: PROFILE
PTIMEBASE =  10 us
Include calls
Status: OK

Event                    : Count :  Time:Time Min:Time Avg:Time Max
-------------------------+-------+-----------+--------+--------+--
:CF6_EN.ENGINE           : 444 :  318 ms:  710 us:  710 us:  720 us
:MPC_HA.MPC_TRANSMIT     : 444 :   41 ms:   90 us:   90 us:  100 us
MPC_RECEIVE_MESSAGE      : 444 :   39 ms:   80 us:   80 us:   90 us
*Background*             : 1331:   22sec:   40 us:   16 ms:   50 ms
-------------------------+-------+-----------+--------+--------+--
      Total              :       2663:   22sec
?
?
```

make the interrupt to routine activation measurents

mpc interrupt to start of the interrupt routine

```
?pat init duration interrupt to :c286_int_handler.mpc_interrupt
?histo = false
?pat display

Mode: DURATION
Event: *INT*-:C286_INT_HANDLER#15
Time Range: 10us TO 1sec

PTIMEBASE = 200 ns
Status: OK

    Time Interval :  Bin Sum : Time    ]
------------------+----------+--------]
        <    10 us :         0:     0 ns]
   10 us-    11 us+:         0:     0 ns]
   11 us-    19 us+:        70:   919 us]
   19 us-    75 us+:         0:     0 ns]
   75 us-   457 us+:         0:     0 ns]
------------------+----------+--------]
Time Min:    12.8 us
Time Max:    13.6 us
```

280691-42

5

```
mpc interrupt to start of mpc_receive_message

?pat init duration interrupt to
:mpc_handler_module.mpc_receive_message
?pat display

Mode: DURATION
Event: *INT*-00109DH
Time Range: 200ns TO 1sec
PTIMEBASE = 200 ns
Status: OK

    Time Interval :  Bin Sum : Time    ]
------------------+----------+--------]
        <  200 ns :         0:     0 ns]
  200 ns- 1200 ns+:         0:     0 ns]
 1400 ns- 9200 ns+:         0:     0 ns]
 9400 ns-   65 us+:        71: 1523 us]
   65 us-  447 us+:         0:     0 ns]
------------------+----------+--------]
Time Min:   21.0 us
Time Max:   22.0 us
?
?
?
?/* end of test session */
?
?halt
*Probe 0 stopped at :MAIN_MODULE_286 + 74H because of halt
  Clips= F6  Trace Buffer Overflow
*exit
 I2ICE terminated
```

280691-43

intel®

AP-433

APPLICATION NOTE

Simple I/O Design Example using the Multibus II Modular Interface eXtension (MIX) Architecture

5

ERIK A. STEEB
OMSO TECHNICAL MARKETING ENGINEER

September 1993

Order Number: 281004-002

Simple I/O Design Example using the Multibus II Modular Interface Extension (MIX) Architecture

CONTENTS

PAGE

1.0 INTRODUCTION 5-175

2.0 MIX APPLICATION EXAMPLE 5-176

2.1 Flight Simulation System 5-176

3.0 MODULE DEFINITION 5-177

3.1 Module Purpose 5-177

3.2 I/O Component Definition 5-177

3.3 Module Functional Blocks 5-177

4.0 DESIGN TOOLS USED 5-178

4.1 OrCAD* Schematic Capture Package 5-178

4.2 PALASM* 2 5-178

5.0 MODULE LOGIC DESIGN 5-179

5.1 Parallel I/O Block 5-179

5.2 Serial I/O Block 5-180

5.3 Data Driver Block 5-180

5.4 MIX Interface Block 5-181

5.5 Module Control Block 5-183

5.5.1 I/O Decode Block 5-184

5.5.2 I/O Control Block 5-185

5.6 Serial EEPROM Block 5-189

5.7 Simulation 5-189

CONTENTS

PAGE

6.0 MODULE FIRMWARE 5-190

6.1 Serial EEPROM 5-190

6.2 BIST Support Levels 5-192

7.0 MODULE PROTOTYPING 5-192

7.1 Electrical Considerations 5-192

7.2 Wire Wrap Layout 5-192

7.3 Module Debug 5-192

8.0 CONCLUSIONS 5-193

APPENDIX A SCHEMATICS 5-194

APPENDIX B MIX MODULE PAL CODE 5-208

1.0 INTRODUCTION

Since its introduction, Multibus II has proven to be a very good solution for high performance computing. The bus is well suited high performance applications due to its ability to support multiple CPUs in a loosely coupled environment. This capability is not only bourne in hardware but in the firmware and software as well.

In this loosely coupled multiprocessing environment, many CPUs are able to interoperate on a common backplane. With this ability comes the need for a system design which is separated along functional boundaries. This means particular I/O needs of the system are handled in a client/server fashion instead of the traditional master/slave method. The difference in the two approaches lies in the ability of the I/O server to support more than one host processor while the slave cannot.

The use of a functionally partitioned system allows for better structured programming and ease of upgrading system resources. However, the designers of special I/O are faced with the challenge of more complex board designs. For Multibus II this means not only designing a board which supports message passing and the multiple address spaces defined for the bus, but also one with an intelligent CPU core and a high speed memory subsystem. These added requirements not only increase the design complexity but can also greatly affect the time to market of a product.

Intel has developed an I/O strategy which eliminates the burden of developing the compute engine and PSB interface of the I/O server, therefore easing the problems associated with designing special I/O for Multibus II. This is done by decoupling the specific I/O technology from the CPU technology in a baseboard/module fashion as shown in Figure 1. With this design, an Intel baseboard is coupled with an I/O module supplied by Intel, the customer, or a third party. Although the baseboard and module are separate boards, the combination forms a single slot I/O server Multibus II.

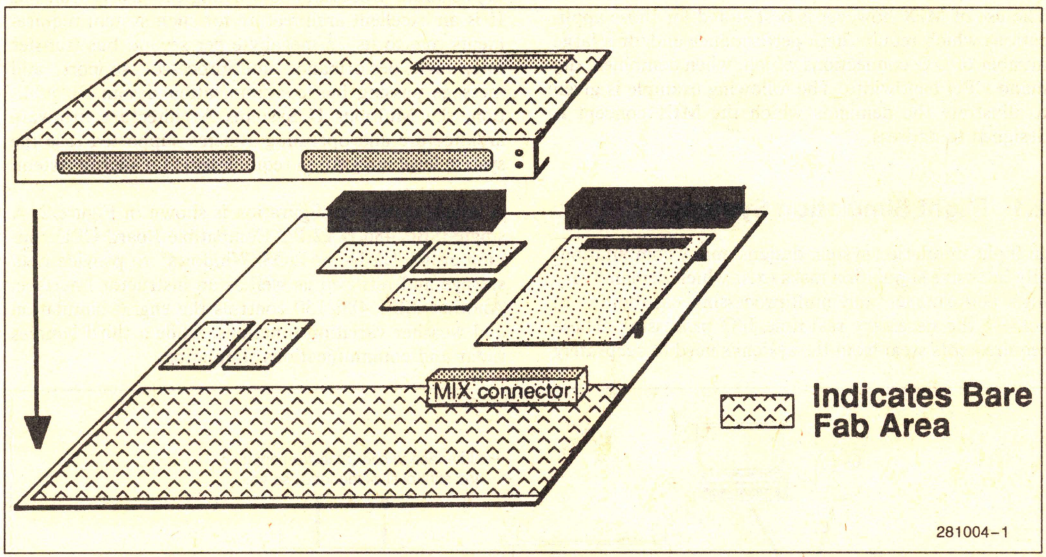


Figure 1. MIX Baseboard/Module Combination

5

MIS is an acronym for Modular Interface eXtension. MIX is a trademark of MIX Software, Inc.
UNIX is a trademark of AT&T, Inc.
OrCAD is a trademark of OrCAD SYSTEMS CORPORATION.
PALASM is a trademark of Monolithic Memories, Inc.

The modular design allows an engineer to focus his efforts on the I/O needs of a product by integrating his special I/O design with an intelligent Multibus II I/O platform supplied by Intel. This platform provides the necessary computational power required of an I/O server but leaves the special I/O module design to the customer.

The strategy outlined above is known as the Modular Interface eXtension (MIX) architecture. This document gives an application example and illustrates a simple I/O design which utilizes the Multibus II MIX architecture.

2.0 MIX APPLICATION EXAMPLE

Because the MIX architecture allows stacking of up to three MIX modules on a high performance baseboard, price/performance scalability is offered along with ease of design for Multibus II, ease of upgrade potential, and support for Intel operating systems. Therefore, a MIX design fits well in virtually any Multibus II application.

The use of MIX however is best suited for those applications which require high performance and/or a large number of I/O connections which, when combined, demand CPU bandwidth. The following example is given to illustrate the demands which the MIX concept is designed to address.

2.1 Flight Simulation System

In flight simulation system designs, many computationally intensive simulation tasks exist which demand very high performance and multiprocessing capabilities to achieve the necessary real-time I/O processing. These requirements stem from the system's need to accurately and quickly convert pilot and aircraft action into control and instrumentation reaction.

This application requires plenty of computational muscle, a fast real-time kernel to handle the simulation tasks, and lots of I/O bandwidth. In addition, a custom interface for cockpit controls and instrumentation is required to handle all aspects of the simulation process.

Consider simulating airflow over a wing. A processor must not only compute the forces on the wing which result from the fluid motion of air over its surface, but also determine and apply (in real-time) the proper amount of feedback pressure to the pilot's control stick.

In addition to the simulation requirements of the system, a flight data collection and storage mechanism, and possibly an instructor interface, will be needed to handle on line customization of the flight variables and post-flight critiques.

These tasks demand multiprocessing capabilities in an environment which allows constant interaction among the separate processors completing the tasks. Multibus II is an excellent architecture for such system requirements due to its 32 megabyte per second bus transfer rate, high performance multiprocessor support, and message passing bus communication design. In addition, the Multibus II Systems Architecture firmware architecture and operating systems supported ease the system level integration requirements for such a system.

A rough system configuration is shown in Figure 2. A single iSBC 486 DX2 PC Compatible Board CPU running either UNIX or DOS/Windows* to provide data storage and retrieval as well as an instructor interface. Another iSBC 486/150 controls the engine simulation and weather variation functions, while a third handles radar and communications simulation.

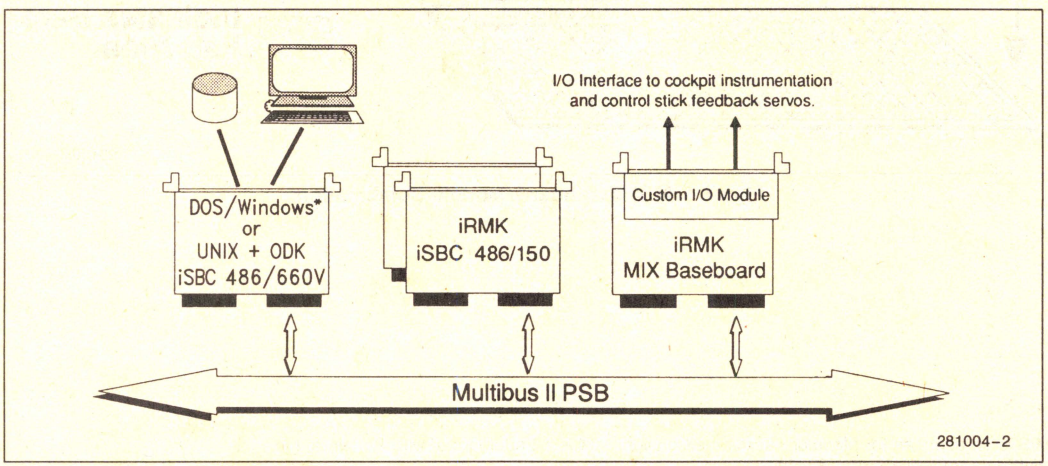


Figure 2. Flight Simulator Example

The MIX baseboard in this example controls the airflow simulation as well as the instrumentation output and control stick feedback I/O. The I/O interface to the actual simulator instruments and control stick is accomplished via a slave MIX module which utilizes a high speed digital I/O link. This module is the only hardware which must be designed by the systems integrator.

In addition to the high performance hardware functions this solution provides, it also offers support for current Intel system firmware and operating systems.

From a firmware point of view, all boards discussed ship with Intel MSA compatible firmware. This allows ease of integration into a system environment in terms of board and system level diagnostics, boot mechanisms, and board-to-board communication support.

Ease of integrating the custom MIX module into this firmware architecture is also provided. Functions in the MIX baseboard firmware allow the I/O module firmware to interact with the baseboard firmware for diagnostics. boot, and operating requirements without the need for modifying the baseboard EPROM contents. This enables any standard or special I/O modules to participate in BIST testing and boot procedures.

The demand for high performance software capabilities can be easily realized with the use of the iRMK realtime kernel. This 32-bit kernel includes the transport and message passing support necessary for the intrasystem communication used in completing the system level task. In this example, the MIX baseboard and each iSBC 486/150 performing simulation tasks runs the iRMK kernel.

UNIX System V/4.2 or DOS/Windows is run on the host iSBC 486 DX2 to provide the operator interface and flight record databasing functions of the system. These functions are typical of the capabilities of standard UNIX or DOS/Windows and exist as software packages supported by the UNIX or DOS/Windows operating system.

With the host running UNIX System V/4.2, the 486 DX2 boards running the iRMK kernel can take advantage of the capabilities of Intel's System V/iRMK. This version of iRMK is designed to operate in a UNIX hosted environment and provides a UNIX hosted development environment as well as a tested transport package for UNIX/iRMK communications. These capabilities greatly simplify the data collection and storage requirements of the system.

The remaining pieces of the system which must be developed by the system integrator are the custom MIX module, the iRMK tasks to perform the simulation, and the instructor interface routines for the UNIX or DOS/Windows subsystem. This document will discuss the MIX module portion of the design.

3.0 MODULE DEFINITION

3.1 Module Purpose

The MIX module to be designed must be capable of supplying many external digital I/O connections which can be quickly accessed and latched for reliable digital I/O performance. The module will also contain a serial interface to the MIX baseboard for debug purposes. These functional goals of the module can be realized in a simple 8-bit slave I/O MIX module design. To improve the aggregate I/O throughput of the design, the module architecture will be extended to support 32-bit accesses to the digital I/O channels.

This module design does not demonstrate the full capabilities of the MIX architecture as the module does not have MIX master capabilities. It does however, function as a useful example to demonstrate the considerations and issues involved in the design of a simple MIX module.

3.2 I/O Component Definition

In this module design example, a simple slave I/O device is used for digital I/O control. The 8255A Programmable Peripheral Interface component provides all of the I/O capabilities required of the external interface. The device provides three 8-bit parallel I/O ports programmable in three modes. The speed requirements of the digital I/O interface are met by utilizing the 82C55A CHMOS version of the chip. This version offers much better timing specifications for command widths and command recovery times as compared to the NMOS 8255A. In addition, the CHMOS device consumes much less power than its NMOS relative.

For the serial interface, the module will utilize the 82510 Asynchronous Serial Controller. This chip is chosen over the more common 82530 due to its proven high performance and its low power, CHMOS characteristics. In addition, this chip is being used on other Intel MIX modules such as the MIX 450 Terminal Concentrator and the MIX 560 Ethernet Controller.

3.3 Module Functional Blocks

The I/O chips chosen above have signal sets based on the 8086 architecture. Because the MIX interface supports an i386™ signal set, the module design must supply a signal conversion function for these components. Furthermore, the module must supply data transceivers, an I/O address decode scheme, and an interconnect EEPROM interface.

By separating the design into functional partitions, the task of transforming MIX signals into signals compatible with the I/O devices becomes a straight forward process.

The features described above are easily separated into five main functional blocks as shown in Figure 3. They are the MIX interface, the data transceivers, the control logic block, the I/O block, and the configuration data block.

The MIX interface block simply consists of the MIX connector and the terminating resistors required for the data bus and control signals. The data buffer block contains the data drivers which provide bidirectional data transfers and support the MIX interface TTL drive requirements. These devices will be set up in a byte swap configuration which allows BYTE, WORD, or DWORD accesses to the I/O devices.

The control logic block is made up of PALs which perform the I/O address decode and convert the MIX bus i386 family signals into read and write commands useable by the parallel and serial components. It also handles the READY logic for the module and generates the proper gate and direction signals for the data drivers. This block is by far the most complex portion of the module. The I/O block contains the parallel and serial chips as well as the drivers and connectors required to interface with external devices. The final block is the configuration data block. This block contains a serial EEPROM device which provides dynamic configurability of the baseboard's interconnect space during power up.

4.0 DESIGN TOOLS USED

Several engineering tools are used in this project to shorten the design cycle, simplify logic design requirements, and provide a means of simulation before the module is prototyped. Because this is a relatively simple design, engineered by a single person, it is desirable to find engineering tools which can run on a common desktop PC. Many PC/AT compatible design tools exist which offer the capabilities needed for this type of board design.

The tools purchased from third parties for this project include a schematic capture package and a PAL design and simulation utility.

4.1 OrCAD Schematic Capture Package

OrCAD* Systems Corporation provides a CAE package which includes a schematic capture design tool and a PCB layout utility. The schematic capture tool is used in this module design.

The OrCAD schematic package offers hierarchical or flat file schematic design, common component libraries, annotation, back annotation, cleanup, cross referencing, error checking, netlist, and parts list utilities. The completeness of this package allows the entire module design to be completed on a simple PC/AT computer.

4.2 PALASM 2 PAL Specification Utility

The control portion of this MIX module must perform address decode functions as well as generate chip selects, provide transceiver gate signals, and handle the READY circuitry. While these functions can be designed using discrete logic, they are more appropriately and efficiently handled using a PAL or EPLD device.

A PAL programming utility is required to input PAL logic equations and convert those equations into fusemaps which a PAL programmer understands. Such a utility is provided by Monolithic Memories, Inc. This utility is called PALASM 2. It is a follow on to MMI's PALASM software which assembles PAL design specification code, generates PAL device fuse patterns in JEDEC format, checks for syntax and assembly errors, and provides a means for simulating the PALs operation based upon given input states.

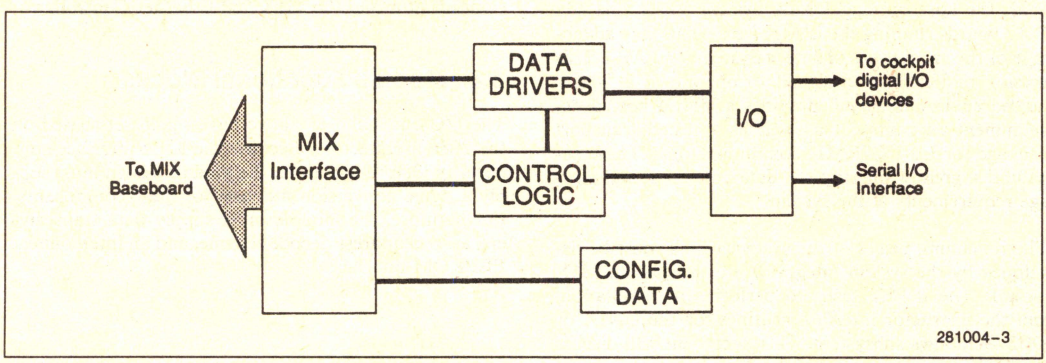


Figure 3. MIX Module Block Diagram

5.0 MODULE LOGIC DESIGN

To complete the logic design for this MIX module, the hierarchical capabilities of the schematic capture package are used in conjunction with the preliminary block diagram shown previously in Figure 3. This procedure allows the design to be broken down into simpler functional design tasks. As shown on sheet 1 of the module schematic (Appendix A), the five functional blocks of Figure 3 have been expanded to eight. A capacitor block has been added to provide the decoupling needed in any TTL design. The MIX block has been broken into the MIX interface block and termination block, and the I/O block has been divided into the Programmable Peripheral Interface (PPI) block and the Serial Interface block.

To start the design with the MIX interface block would not be productive as we have not yet defined the MIX signals which must be supported. Because this design is a MIX slave, the module need only support the MIX signals required to complete its I/O function. To understand which signals are required for this module, the module is designed from the I/O devices back to the MIX interface.

As mentioned earlier, the I/O block defined in Chapter 3 has been divided into two blocks. This is done because the module utilizes two distinctly different I/O devices. We therefore divide the I/O block into a parallel I/O block and a serial I/O block.

5.1 Parallel I/O Block

The 82C55A PPI will serve as the central I/O device for this block. This 8-bit device provides 24 programmable I/O pins which may be individually programmed in 2 groups of 12. It allows direct bit set/reset capabilities and is fully TTL compatible.

As shown in Figure 4, the 82C55A requires 8086 compatible signals for read and write control. The conversion from 386 to 8086 compatible signals as well as the chip select generation will be handled in the Control Block of the design. Aside from RESET, the two remaining chip control signals are A0 and A1. These will be obtained from the MIX interface signals MXA2* and MXA3* respectively. This addressing scheme will have the effect of separating the internal chip registers by 4 bytes.

As stated in section 3.1, the MIX module is to provide a 32-bit data path to the parallel I/O interface. To accomplish this, four 82C55A components operate in parallel. Except for the select lines, the control signals for each chip may be wire ORed together. This is shown on sheets 8 and 9 of the module schematic (Appendix A). To facilitate BYTE, WORD, and DWORD accesses to the digital I/O interface, the data drivers in the Data Buffer Block are designed in a byte swap configuration.

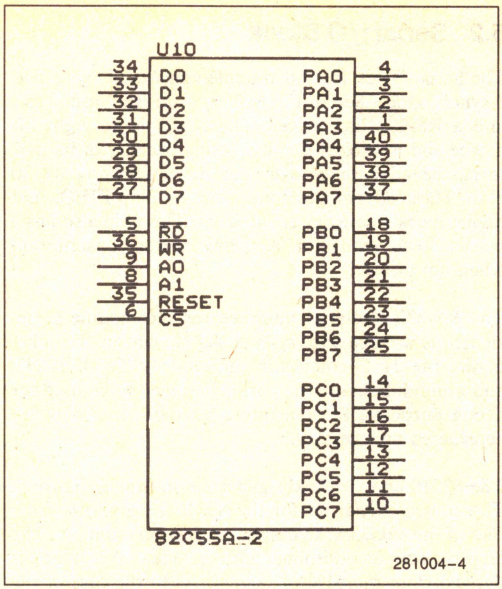


Figure 4. 82C55A PPI

Finally, the Control Block decodes the type of access being made and generates the proper chip select(s) to the peripheral chips.

Because the PPI is capable of only 2.5 mA of DC drive, the parallel I/O block also incorporates line drivers between the PPI and the target I/O devices. In this design, only ports A and C of the PPI are used. This gives 64 bits of digital I/O and uses up to 16 line driver components.

As shown on the module schematic, four 74F245 Octal Transceivers drive the port A signals. Two signals from port B enable and configure the drive direction for the transceivers. These fast devices provide 64 mA of low level output current on the "B" side of the driver. This level is sufficient to drive most digital I/O needs.

Eight 14 pin DIP sockets are used for installation of the port C signal drivers. This allows the user to configure port C with a variety of Quad Two-Input drivers or terminators. This design also uses a 64 pin DIN connector to provide external connection to the 32 port A signals and their associated ground lines. Eight 74ALS00 devices drive the port C outputs to illuminate four banks of eight LEDs. This is shown on the module schematic.

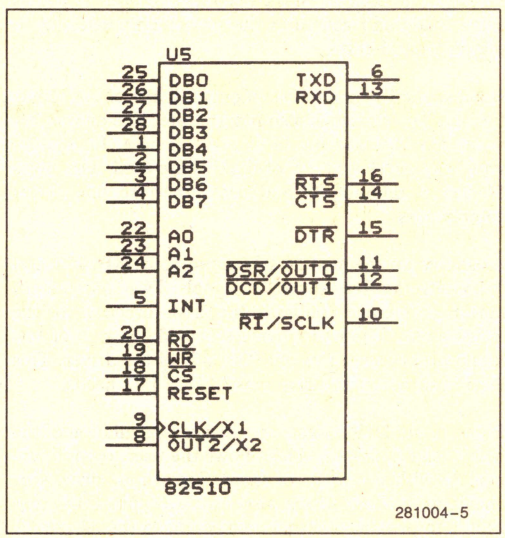
Although Port C of the parallel I/O block in this design is set up for display (LEDs), the block can be easily modified to provide 64 or 96 (using port B) lines of external I/O.

5.2 Serial I/O Block

The Serial Block for this module consists of the 82510 Asynchronous Serial Controller, a crystal oscillator, and a RS-232 line driver. The 82510 is an eight bit device and is chosen for this design because of its performance advantages over the more common 82530 chip. These advantages come from on chip FIFOs and higher clock frequency capabilities. This chip also has a power down mode for very low power consumption when not in use.

The RS-232 electrical interface for the module is designed as a three wire circuit. As shown on the schematic, the J1 D-connector carries only TXD, RXD, and ground. Because the port is designed to be used for debug purposes, the remaining six modem signals are considered unnecessary.

The 82510 and required signals are shown in Figure 5. The control signal set for the 82510 differs little from that of the 82C55A, so design integration is simple. The chip requires an additional address signal, A2, which is derived from the MXA4* signal. As in the case of the 82C55A, the address line connections cause the internal registers of the component to be DWORD aligned.



One interrupt line also comes from the 82510 to signal FIFO conditions, errors, etc. This interrupt line is routed through an inverter and a series terminating resister to the LCLINT* pin of the MIX interface since this is the only interrupt on the module. The MIX interface termination requirements are found in the *MIX Module Design Specification* (Intel p/n 500729-002).

The chip also requires a clock source for baud rate generation. The 82510 allows two modes of operation for the clock source. The first mode uses an external oscillator to internally generate its system clock while the second uses an externally generated clock. This module design opts for the former for several reasons. First, an internally generated clock has lower cost and is electrically "cleaner" than using an external clock. Furthermore, the low power sleep mode of the chip is easily utilized with an internally generated clock. This mode requires the RTS pin to be driven low during a reset. This is provided by the CLKMOD signal shown on the schematic. In this mode, we will use an external 18.432 MHz crystal which results in a 9.216 MHz internal clock (/2). This crystal frequency provides a 0% error baud rate generation for standard baud rate values.

The serial driver used is an MC145406 component. This device is RS-232 compatible and supports both transmit and receive signal voltage conversions on a single chip. Because only the Transmit Data and Receive Data signals are driven, a single MC145406 is sufficient for this application.

Now that the I/O blocks have been specified and the data line requirements known, we may proceed to design the data driver block of the MIX module.

5.3 Data Driver Block

As stated in the design requirements, the module must support 32-bit accesses to the four PPI devices as well as eight bit accesses to the serial device. Restricting parallel port accesses to only 32 bits would limit the baseboard software to 32-bit code or limit the number of PPI devices accessible with BYTE or WORD transfers. To solve this problem, each PPI will be accessible as a separate I/O device or all PPIs will be accessible as a single 32-bit I/O device.

This structure requires a byte swap mechanism in the data driver design. The design allows each PPI and the 82510 ASC to be accessed on the data signals MXD0–MXD7. Furthermore, to obtain a "32-bit" parallel port, one PPI (PPI1) is accessible on MXD8–MXD15, one (PPI2) on MXD16–MXD23, and one (PPI3) on MXD24–MXD31. This configuration requires 7 data transceivers as shown on the schematic.

When accessing the four PPIs as a single 32-bit device, data transceivers U7, U9, U19, and U29 must be enabled at the same time. When accessing the PPIs as separate 8-bit devices, the U7, U8, U18, or U28 transceiver is individually enabled to provide a connection between the PPI being accessed and data signals MXD0–MXD7.

The "G" pin of the transceiver stands for "gate" and is used to enable the driver. The data driver setup requires 5 gate signals to enable the proper transceivers for a PPI transfer cycle. The signal GAT32* is activated during a 32-bit parallel port access. GATB0* transfers data between the U10 PPI and the low data byte on the MIX interface, GATB1* does the same for the U20 PPI, and so on. These signals, along with the data direction signal, are generated by the Control Logic Block of the module design.

5.4 MIX Interface Block

As stated previously, a MIX slave module need only support those MIX interface signals which allow it to complete its I/O function and remain MIX compatible. Now that the I/O device and data drivers blocks have been defined, we can determine the MIX interface signals needed to support this simple slave design.

The 130 signals on the MIX interface can be broken up and viewed as 11 functional groups. This eases the task of selecting the necessary signals. Aside from the Spares group, the MIX slave module signal groups are listed in Table 1. Notice that a slave module does not utilize any of the signals of the Arbitration Group as it will never need control of the interface.

The address group consists of 30 address lines, four byte enables, and three select lines. Because the module is a simple slave without memory mapped devices, only I/O address space need be considered. A MIX module is allotted 1 Kbytes of I/O address space in either the

Table 1. MIX Slave Module Signal Set

Function	Signal Name	Signal Type	Used
Address	MXA[31:28]*	Pass-through	N
	MXA[27:7]*	Bussed	N
	MXA[6:2]*	Bussed	Y
	MXBE[3:0]*	Bussed	Y
	LCLSEL*	Owned	Y
	MXSEL[2:1]*	Pass-through	N
Data	MXD[31:0]	Bussed	Y
Status	MXMIO	Bussed	Y
	MXWR	Bussed	Y
	MXDC	Bussed	Y
Control	MXCYC*	Bussed	Y
	MXCMD*	Bussed	Y
	MXBS16*	Bussed	N
	MXWAIT*	Bussed	Y
Configuration	LCLID*	Owned	Y
	MXID[2:1]*	Pass-through	N
	SCLK	Bussed	Y
	SDIN	Bussed	Y
	SDOUT	Bussed	Y
	MXRST*	Bussed	Y
Interrupt	LCLINT*	Owned	Y
	MXINT[2:1]*	Pass-through	N
DMA	LCLDRQ*	Owned	N
	MXDRQ[2:1]*	Pass-through	N
	LCLDACK*	Owned	N
	MXDACK[2:1]*	Pass-through	N
Option	LCLOP*	Owned	N
	MXOP[2:1]*	Pass-through	N
Spares	Reserved	Bussed	N
Power	+5 VDC	Bussed	Y
	GND	Bussed	Y
	+12 VDC	Bussed	Y
	−12 VDC	Bussed	Y

second, third, or fourth Kbyte in the baseboard's I/O space, depending on the stack position of the MIX module. Luckily, the baseboard's decode logic determines which module is selected and signals that module with a MXSELx* signal, seen as LCLSEL* on the module. This way, the baseboard assigns a module's base I/O address at 400H, 800H, or 0C00H and a module is designed based only on address offsets within that 1 Kbytes of space. In this scheme however, the module must use the LCLSEL* signal as a qualifier in the address decode scheme to assure it is the module being accessed. To support 1 Kbytes of address space, we need only support MXA2* through MXA9* and the byte enable signals. The module must also input its LCLSEL* signal to aid in the decode scheme. Furthermore, although the module does not act on the remaining address and MXSELx* signals, it must route the signals from the primary to secondary sides of the module. This is true for all signals of the MIX interface.

Because this module has been defined to support 32-bit accesses to the parallel I/O devices, all 32 data lines must be supported. All signals in the status group must also be supported to determine the type of access being made to the module.

In the Control group, the signals MXCYC* and MXCMD* are used to determine the state of a MIX transfer cycle. These two signals must be supported by even the simplest of modules. These signals are very useful in generating I/O commands to devices as well as chip select signals. The other signals in the Control group, MXBS16* and MXWAIT*, need only be used by modules which cannot support all 32-bit byte enable combinations or cannot accommodate full speed MIX transfers.

The MXBS16* is used to inform the baseboard that the 32-bit access being made is not supported by the module. This is the same as the BS16* signal for the i386 processor. This MIX module will support 32-bit access to the four PPI device group but not the individual PPIs or 82510 ASC. Therefore, the MXBS16* signal is generated for any access made on the module except an access to the PPI0 address since this is the address we use to enable all four PPIs during a 32-bit access.

To determine the need for the MXWAIT* signal, we must investigate the requirements of the I/O devices being used on the module and understand the result of using the MXWAIT* signal. The MXWAIT* signal can only be used on the MIX interface to extend the cycle time of the MIX transfer being made (i.e., extend the length of the MXCMD* and MXCYC* active time). This signal is very similar to the iSBXMWAIT* signal. Because of the effect of this signal, we need only consider the command duration time requirements of the device. Timing needs for parameters such as address or data setup, hold, and command recovery timings are not affected by simply activating the MXWAIT* signal. Meeting these requirements will necessitate additional control circuitry between the MIX interface and the I/O device.

Timing requirements for TTL components, of course, vary with the device. Typically, as is the case in this module design example, it is the peripheral devices which have the longest timing requirements. A full speed MIX transfer cycle is specified as 175 ns which corresponds to a 73 ns MXCMD* pulse width. If we use the MXCMD* signal to generate our RD* and WT* commands, we would have a minimum command pulse of 73 ns. Although the current MIX baseboard does not support a full speed MIX transfer, it is desir-

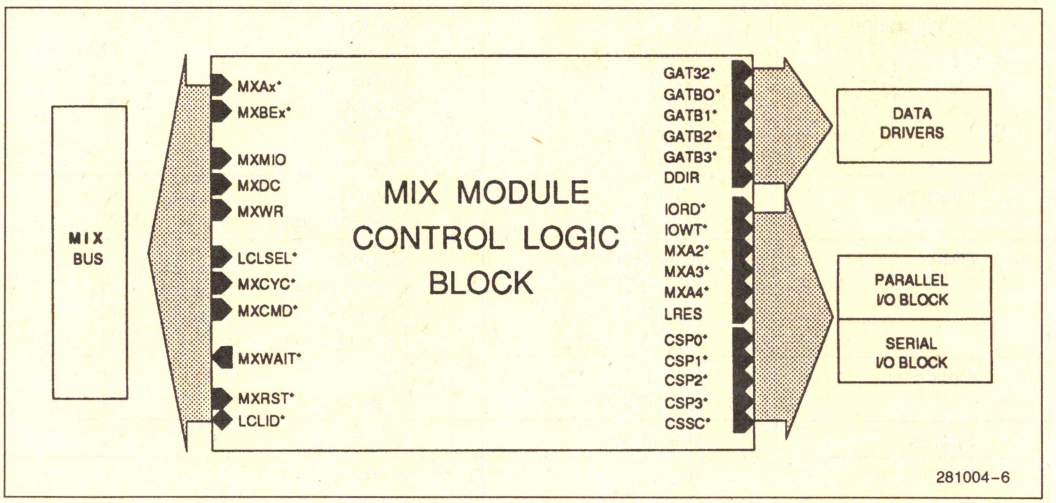


Figure 6. Mix Module Control Block

able to design the MIX module to the capabilities of the MIX specification (not the baseboard spec) so that the module will be compatible with future MIX baseboards. For this reason, the decision to support MXWAIT* is based upon a 73 ns command pulse time.

In the case of the PPI and the 82510 ASC, the parameters we are concerned with are the I/O Read and Write command hold times. The 82510 requires a minimum RD* active width of 2 T_{CY} + 65 ns which is 282 ns. Already we see we cannot support a full speed MIX transfer and will therefore need to use the MXWAIT* signal.

Support of all signals in the configuration group is a requirement of the MIX specification. These signals allow the baseboard and other PSB agents to determine the types of MIX modules present in any given MIX stack.

The Interrupt group of the MIX interface consists only of a single interrupt line per module. This interrupt line supports cascading which facilitates the use of a slave interrupt controller on a MIX module. Because this module has only one interrupt signal source (the serial controller), the cascading option is not needed. The remaining signal groups include the DMA and Option lines. This module does not utilize baseboard DMA resources nor does it require an additional interrupt signal, so these signals need not be supported. As spares go, MIX modules are required to bus the signals for future use but may not drive the signals.

It is a requirement of the MIX specification to connect to all power pins of any voltage which is used on the module. Because this module supports RS-232 communications, all available voltage levels are utilized. The module must therefore connect to all +5V, +12V, −12V, and GND pins.

As shown in Table 1, the final summary of supported signals for this module is designated with a "Y" in the "Used" column. Now that the MIX interface signal support and I/O blocks are defined, the next task is to design the block which converts the MIX compatible signals to commands understood by the I/O devices being used. This is accomplished in the Control Logic Block of the Module.

5.5 Module Control Logic Block

As shown in Figure 6, the MIX module control logic block resides between the MIX interface and the module's I/O and data driver devices. Because this block is similar to that which may be utilized on other MIX slave modules, detailed attention will be given to its design.

The control block has three basic functions; translating MIX signals into commands and timings understood by the I/O devices, performing address decode operations, and handling the module reset functions. The reset function is the simplest of the two, so it will be designed first. The reset function on the MIX interface is driven by the baseboard's microcontroller and, as stated in the module design specification, is guaranteed to be a minimum of 10 μs in duration. The requirements of all devices used in this module design are at least an order of magnitude less than this. For this reason, we need only invert the MXRST* signal to supply the active high RESET required by the peripheral devices.

Another requirement of MIX modules is the support of the MXIDx*/MXEECSx* signal. During reset, the module must drive its LCLID* signal to inform the baseboard it is present. We can generate this signal by simply inverting the inverted MXRST* signal during a reset (this is accomplished with a tri-state inverter which drives the signal only during reset). Note that we could not simply connect the MXRST* to the LCLID* signal, or simply invert the MXRST* signal at all times since LCLID* takes on the function of LCLEECS* after reset. If we had simply tied MXRST* to LCLID*, an access to the serial EEPROM would result in a MXRST* being active on the MIX bus. If we did not tri-state the LCLID* driver, the baseboard MXEECSx* driver and our LCLID* driver would attempt to drive the same signal to opposite polarities. Also note that this design meets all reset timing requirements as listed in the *MIX Module Design Specification*.

The address decode functions and command conversion logic are the next portions of the control block which must be designed. Because of the relative isolation between these two functions, they can be partitioned into two designs, however, proper timing between the two functions must be ensured.

Typical operation for a peripheral chip is to set up address and chip select signals to the device, set up the data (for a write), and then pulse the read or write command. To do this, we can construct two logic subblocks we will call the I/O control and I/O decode blocks.

The I/O control block will be responsible for generating the RD* and WT* pulses to the chips, and controlling the MXWAIT* and MXBS16* functions.

The I/O decode block therefore has the responsibility of performing all the address decode operations and generation of the chip selects and data driver gate signals. These functions are derived from the states of the address and byte enable signals during a valid transfer cycle. The functions in both blocks are best handled by PAL or PLD logic design due to the device's logic flexibility and the number of input signals required of the functions.

5.5.1 I/O DECODE PAL

The main function of this block is the generation of the peripheral device chip select signals. Each addressable device on the module is assigned a unique I/O offset address by this PAL. A device's chip select is only activated when the MIX baseboard accesses the chip's assigned I/O address via the MXA[6:2]* and LCLSEL* signals.

In addition to a valid address, it is wise to ensure the proper byte enable combination exists to provide valid data to the chip being selected. Therefore, our chip selects will be qualified by three things; one, a valid MIX transfer cycle is present; two, the proper I/O address is selected; and three, a valid byte enable combination exists. Because each of the peripheral devices require MXA3* and MXA2* for internal register access, we need not use these in qualifying accesses to a specific chip. This will force our address decode to place the peripheral devices on DWORD boundaries.

The addresses for each chip may be arbitrarily chosen between I/O offsets 0 and 3F0H. Note that as specified in the *MIX Module Design Specification*, I/O address offsets 3F0H through 3FEH are reserved. This module utilizes address lines MXA[6:2]* for its decode function. Because this module does not decode address lines nine through seven, the I/O chips are mirrored in the address map every 128 bytes. The first occurance of the chips I/O addresses are shown in Table 2.

Table 2. I/O Addresses

I/O Offset	Device
0xH	Byte PPI (PPI0) Word PPI (PPIs 0,1) Dword PPI (PPI 0–3)
1xH	Byte PPI (PPI1)
2xH	Byte PPI (PPI2)
3xH	Byte PPI (PPI3)
4xH	Byte 82510 ASC
5xH	

A secondary function of this block is the control of the data transceiver gate signals.

There are seven data transceivers in the Data Driver Block of the module which have enable pins ("G") which must be driven during peripheral accesses.

Of the seven data drivers three are only used for 32-bit transfers. These may be hard wired together and enabled as a group during a 32-bit access (This signal will be called GAT32*). The four remaining gate signals are used during byte wide accesses to the PPIs (We will call these GATB[3:0]* based on which byte of the module data bus the driver controls).

The data driver gate signal /GATB0 enables the data driver which transfers data from the low byte of the MIX data bus to the low byte of the module's data bus. PPI0 and the 82510 reside on this bus. This driver should be enabled on four access occasions. Three involve byte, word, or dword accesses to PPI0 and the fourth is for a byte access to the serial chip. The /GAT32 signal enables three data drivers which transfer data from the upper three bytes of the MIX data bus to the corresponding upper three bytes of the module's data bus. This signal should be activated only during word or dword accesses to PPI0. By providing simultaneous chip selects to the other PPIs, we can make the baseboard believe it is making an access to a single word or dword device when in actuality, two or four PPIs are being utilized.

The remaining gate signals (GATB[3:1]*) allow the upper three PPI devices to be accessed via the low byte of the MIX data bus. These three signals drive data between the low byte of the MIX data bus and one of the upper MIX module data bus bytes (1, 2 or 3). The signals are only activated during byte accesses to PPIs 1, 2 or 3.

Because the data timing for our devices must conform to the MIX interface timing requirements as well as the timing requirements of the peripheral chips, we must activate these drivers in a manner which meets the minimum data turn-on and minimum data hold time requirements. This can be accomplished by dividing the enable function between read and write accesses to the module. During a read operation, the module must tristate its data drivers within 40 ns of the MXCMD* signal being deactivated. During write accesses, the 82C55A requires data be held at least 30 ns after the RD* or WT* signal is removed. Both needs are met by varying the gate parameters based on whether a read or write access is occurring.

From these definitions we find the total number of signals used by the decode logic is 12 inputs and 10 outputs. A common 22V10 PAL facilitates this logic design perfectly.

The PAL signal definitions are shown in Table 3. Note that the chip select generation scheme for the PPI devices will also allow us to make 16-bit accesses to the two PPIs which occupy the low word of the data bus (U10 and U20) without modification to the data driver scheme.

The actual PAL source code for this block is listed in Appendix B under the title "IODCD PAL".

Table 3. I/O Decode PAL Signals

Signal	Description
/A6–/A4	Inputs from MXA9*–MXA4*
/BE3–/BE0	Inputs from MXBE3*–MXBE0*
/LCLSEL	MIX i/f input
/MXCMD	MIX i/f input
/MXCYC	MIX i/f input
MXWR	MIX i/f input
/GATB0	Byte 0 data driver gate signal
/GATB1	Byte 1 data driver gate signal
/GATB2	Byte 2 data driver gate signal
/GATB3	Byte 3 data driver gate signal
/GAT32	Bytes 1, 2 and 3 drivers gate
/CSP3	PPI3 chip select
/CSP2	PPI2 chip select
/CSP1	PPI1 chip select
/CSP0	PPI0 chip select
/CSSC	82510 chip select

5.5.2 I/O CONTROL PAL

As discussed earlier, the I/O control portion of the design is responsible for translating MIX compatible signals into commands recognized by the peripheral devices. It must also provide the MXWAIT* support for proper command timing and MXBS16* support for proper data accesses to the peripherals.

Since the devices supported on this module will act only on I/O read and I/O write commands, this PAL must translate the MIX Status group signal states into either IORD* or IOWT* commands to the 82510 and PPI devices. This is accomplished by decoding the states of MXMIO, MXDC, MXWR, LCLSEL*, MXCYC*, and MXCMD*. As an added check, we will qualify our decodes with the state of the MXRST* signal to ensure chip selects are not generated from the arbitrary signal states present during reset.

Because of the pulse width and command recovery timing requirements of the 82510 and the 82C55A we must support the MXWAIT* signal of the MIX interface. This signal will directly fulfill our pulse width needs and indirectly provide the command recovery time needed by the peripheral devices. As stated in the *MIX Module Design Specification,* the MXWAIT* signal must be asserted within 40 ns of the MXCYC* signal being activated. This requirement will require asynchronous and synchronous state machines to be linked in the PAL logic.

The I/O Control Block will also drive the MXBS16* signal on the MIX interface. A MIX module must drive this signal when it cannot support 32-bit accesses to its devices. This module design will activate this signal on all accesses except for those to the PPI0 address since this address is the one used for our 32-bit PPI access. To distinguish a PPI0 access, the control PAL will input several of the address lines.

An additional function not yet mentioned is the direction signal generation required by the data transceivers. Because these transceivers are only enabled during valid command cycles, we simply drive the direction pins with the MXWR signal. Note that the stub length requirements of the MIX interface and the drive requirements of the transceivers necessitate the use of a driver to produce the direction signal. This driver is a simple hex noninverter and may be implemented external to the PAL design.

As specified, our PAL design will require twelve inputs (including a clock signal for the MXWAIT* machine) and four outputs. From these requirements a 22V10PAL device may again be used. This device allows twelve input and ten output pins. Although we have only defined four outputs, additional output pins will be consumed by the WAIT state machine design. A particularly nice feature of the 22V10 device is the ability to configure the outputs as either registered or simple combinatorial output pins. This is useful since we have both asynchronous and synchronous needs for this PAL design.

The first consideration in our PAL design is the tri-state needs of the MIX bus signals. The MIX specification dictates that the MXWAIT* and MXBS16* signals be tri-stated while a MIX cycle is not occurring. Furthermore, these signals must be in their logical false state before they are tri-stated. To facilitate this, we define a signal called IOCYC.* This signal is a combination of inputs which represent a valid I/O cycle (read or write) is occurring. The actual input signals can then be used to activate and deactivate the MXWAIT* and MXBS16* signals, while the IOCYC output is used to tri-state them. This provides 15 ns (the prop delay of the PAL) between the signals being deactivated and being tri-stated.

The IORD* and IOWT* signals can build upon the IOCYC* signal by using IOCYC* as a qualifier. Once IOCYC* is valid, the MXWR signal is used to distinguish between read and write accesses, and the MXCMD* signal is used to allow address and chip select setup times to the peripheral devices. The read and write commands will require one additional qualifier to provide the command recovery timing needs for the PPI chips. This addition will be discussed during the WAIT state machine design.

The MXBS16* function is a fairly simple signal to generate for this design. The module will only allow 32-bit accesses to the address specified for PPI0. This address, along with the Byte Enable signals, is used in the decode logic to enable all PPIs simultaneously when a 32-bit access is made to PPI0. Therefore, this PAL will not drive the MXBS16* signal during an access to PPI0. This is assured by verifying an access to PPI0 is not occurring.

Because of the propagation delays involved in PAL decode, we are not able to use the chip selects generated by the I/O Decode PAL for the generation of MXBS16*. The 15 ns delay of the chip select would result in a MXBS16* glitch during a PPI0 access. We can however, use the address lines MXA[6:4]* as signal qualifiers. These signals are specified to be glitch free during valid transer cycles and will indicate which device is selected when MXCYC* is activated. We therefore use the address lines with MXCYC* line in generating MXBS16*. Note that if the PAL offered more inputs, the signals MXA[9:7]* could also be used which would cover the full 1 kilobyte address range of the module.

The last requirement placed on the MXBS16* signal is that it be tri-stated within 46 ns of MXCYC* going inactive. This is so because the MXBS16* signal is bussed on the MIX interface. Since our 22V10 PAL device allows us to individually assign tri-state conditions to outputs, we can use IOCYC* to tri-state the MXBS16* output. This allows us to force MXBS16* inactive (high) with the removal of the MXCYC* signal and tri-state the line 15 ns (the propagation delay of the device) later when IOCYC* is deactivated.

The final function provided by this PAL is the MXWAIT* support. As discussed previously, the 82510 device requires a command pulse width of 282 ns for a read operation and 232 ns for a write. The 82C55A PPI device requires pulse widths of only 150 ns for a RD and 100 ns for a WR. Although we could include logic to distinguish PPI accesses from 82510 accesses, a generic MXWAIT* which supports the worst case time is much simpler to design. Therefore, a state machine must be designed which will produce a pulse greater than or equal to our maximum pulse width constraint of 282 ns.

Our peripheral devices also specify command recovery requirements which are not met by a full speed MIX cycle. The command recovery is the amount of inactive time between RD* or WT* pulses to single chip. For the 82510 this value is 120 ns. The PPI requires 200 ns. As stated in the *MIX Module Design Specification,* all modules MUST provide hardware support for command recovery needs. This support can be rolled into the MXWAIT* state machine by designing in a number of inactive states in which the RD* or WT* signals cannot be activated. To assure these times are met and to assure we meet the MXWAIT* activation time requirements, we will utilize an asynchronous state machine to generate the actual MXWAIT* signal, coupled with a synchronous state machine which "turns off" the asynchronous machine. We will also use the output of our synchronous machine as a qualifier in the IORD* and IOWT* signal generation. This will give us the command recovery time needed for proper operation of the 82510 and 82C55A chips.

The asynchronous machine will define the conditions which must be present for the MXWAIT* signal to be asserted. In the synchronous state machine design, we must define three things; the clock frequency for the PAL device, the number of states in which SWAIT (synchronous WAIT) is active, and the number of states SWAIT is inactive. By qualifying IORD* and IOWT* with an active SWAIT, the added inactive SWAIT states provide our recovery time.

For the asynchronous machine, we define an idle state for a reset state or otherwise idle module (no transfer occurring) condition. Once the module is accessed and MXCYC* is asserted, we must provide MXWAIT* within 40 ns. A second state is therefore defined with MXWAIT* active. This state is entered when IOCYC* is active. With the PAL device used, we know MXWAIT* will be activated a maximum of 30 ns after MXCYC* is activated (15 ns from MXCYC* to IOCYC* and 15 ns from IOCYC* to MXWAIT* coming out of tri-state). This second state is held until the synchronous machine sets SWAIT active. Once this occurs, the machine moves into a third state which also has MXWAIT* active. This state is held until the synchronous machine indicates the delay time has expired (falling edge of SWAIT). The asynchronous machine then deactivates MXWAIT* and enters its final state until the cycle has completed. This state exists so that the loop is not repeated during the same MIX transfer cycle. Once IOCYC* goes inactive, the idle state is reentered.

To design the synchronous machine, we need to know the requirements for pulse widths and recovery times. As stated previously, our longest pulse width requirement comes from an 82510 read cycle (282 ns) and our worst case recovery time requirement comes from back to back 82C55A accesses (200 ns). With these in mind, the equation $n/f = T_{WAIT}$ is used to determine the

number of active and inactive WAIT cycles, "n", used in our synchronous state machine. The value "n" will vary based on the clock frequency, "f", of the PAL, and the required cycle time, "T_{WAIT}".

For the active cycles, T_{WAIT} represents the maximum command pulse width. For the inactive cycles, T_{WAIT} is the maximum command recovery time. Since both of these values are divisible by 40 ns we will use an "f" of 25 MHz (1/25 MHz = 40 ns). For this frequency we need 7 active states and 5 inactive states. Remember these active and inactive states represent states where MXCMD* is active or inactive respectively. We must therefore add an additional state for both the active and inactive transitions which waits for MXCMD* to transition to its desired state.

Two state diagrams are shown in Figure 7. The top represents the synchronous machine while the bottom represents the asynchronous machine. For the synchronous machine, four state variables are defined: S0, S1, S2, and SWAIT. The "0000" state is the machine's idle state. Once a cycle is initiated on the module and the asynchronous machine asserts MXWAIT*, the synchronous machine moves through its states as shown in the diagram. Notice that the 0000, 0001, and 1000 states require MXWAIT* or MXCMD* to transition before the next state is entered. The MXWAIT* requirement ensures the cycle is only entered once per valid module transfer cycle. This is accomplished without requiring the synchronous machine to be in a specific state when a MIX cycle starts. Note that a new transfer cycle could start anywhere between the 1100 and 0000 states of the synchronous machine. The MXCMD* loops exist to ensure the command widths and recovery timing needs are met. Note that SWAIT is asserted with MXWAIT* active but the clocked cycle is not executed until MXCMD* is activated. This is desirable since MXCMD* is the signal we need to extend by our T_{WAIT} value. Similarly, once SWAIT is deactivated, the machine will wait for MXCMD* to be deactivated before continuing with its cycle. By only allowing the IORD* and IOWT* commands to be activated while SWAIT is active, we satisfy our recovery time requirements by assuring SWAIT is inactive for our second (recovery time) T_{WAIT} value after MXCMD* is released.

SWAIT is the synchronous variable which is linked with the asynchronous machine. Regardless of the state of SWAIT, MXWAIT* will be activated with the onset of a valid MIX cycle (IOCYC* active). Once activated however, MXWAIT* will remain active until the falling edge of SWAIT. Once deactivated, MXWAIT* will not be reactivated until the falling edge of IOCYC* is detected.

With this machine, MXWAIT* will always be activated within 40 ns of MXCYC* being activated, however, the IORD* or IOWT* command will not be activated until recovery requirements from the previous access are met (SWAIT goes active).

Remember that the MXWAIT* signal is tri-stated when IOCYC* is inactive. This is necessary because of the bussed nature of the MXWAIT* signal and is accomplished in the PAL specification with the IOCYC* signal. The requirement that MXWAIT* be in its inactive (HIGH) state before it is tri-stated is also met with this design because MXWAIT* will go inactive while both MXCMD* and MXCYC* are asserted. MXWAIT* is then tri-stated by the IOCYC* signal going inactive.

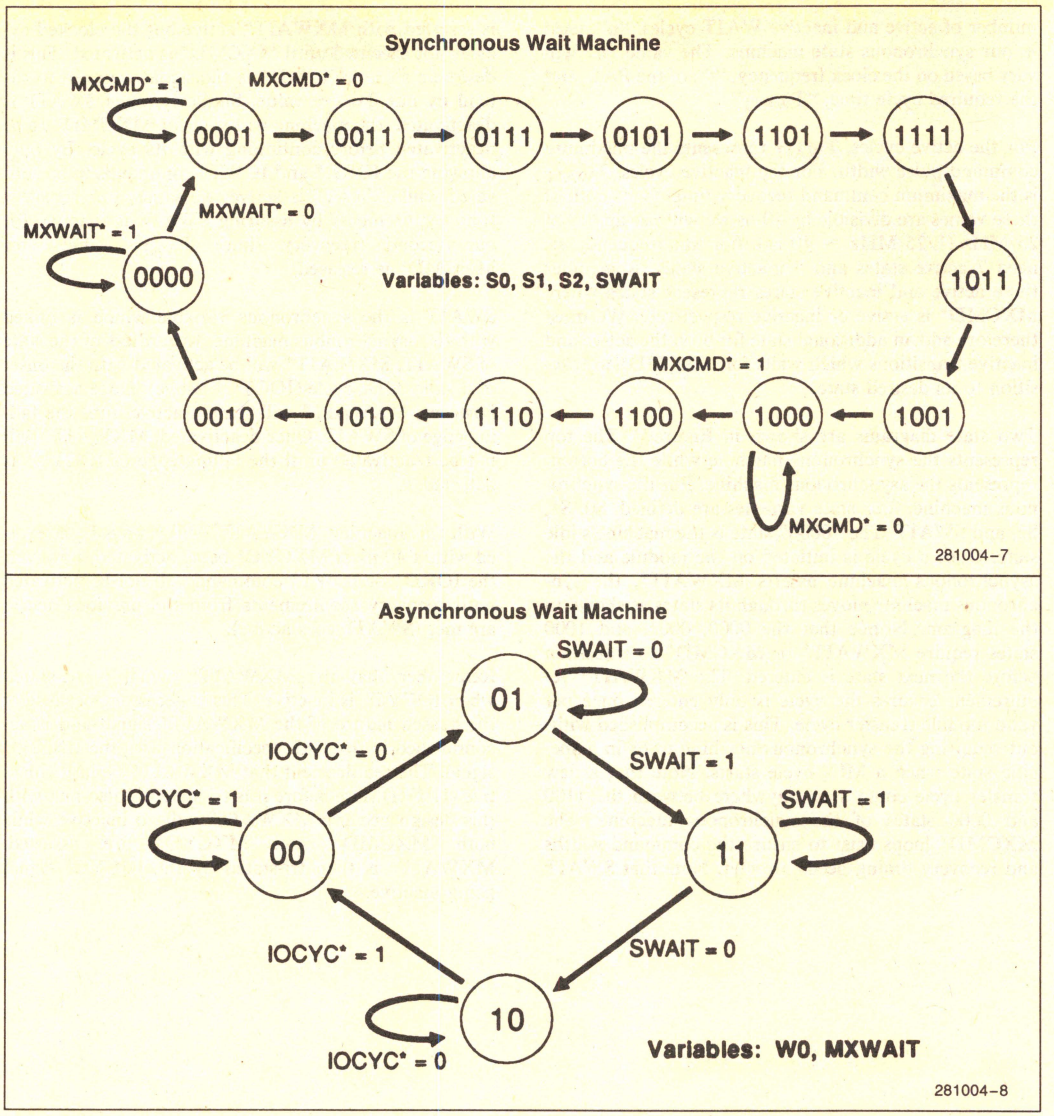


Figure 7. MXWAIT Machine State Diagram

Although the W0 and Sx variables are only used as internal variables to the PAL device, they must be assigned to actual pins on the 22V10PAL device, bringing the number of output pins on the device to ten. Furthermore, the Sx and SWAIT variables are defined in the PAL source as registered (synchronous) outputs via the " := " syntax.

From the asynchronous machine states, we can assign a truth table as shown in Table 4. Note that Table 4 gives values for CYC which is the logical state of the IOCYC* signal (IOCYC* = 0–CYC = 1). Also "SW" represents the state of SWAIT and WAIT represents the logical value of MXWAIT (i.e. WAIT = 1–MXWAIT* = 0). This leads to the equations for W0 and MXWAIT as listed in the PAL source code in Appendix B.

The synchronous machine follows a similar design, although in the synchronous case there are six variables instead of four. The PAL source for the synchronous machine is also listed in Appendix B.

Table 4. WAIT State Machine Truth Tables

Current State				Next State	
W0	WAIT	CYC	SW	W0	WAIT
0	0	0	0	0	0
0	0	1	0	0	1
0	0	1	1	0	1
0	1	1	0	0	1
0	1	1	1	1	1
1	1	1	1	1	1
1	1	1	0	1	0
1	0	1	0	1	0
1	0	0	0	0	0

For the I/O Control PAL's 25 MHz clock source, a 50 MHz crystal oscillator is routed through a D flip-flop which provides a divide by 2 function. The output of the flip-flop is then tied to the clock input pin of the PAL.

The final piece of the I/O control block which must be designed is the logic which supplies the address lines to the peripheral devices. Because the devices require active high address lines, we need only invert the MXA4* through MXA2* signals and route them appropriately. Note that the 82510 is the only device which requires the MXA4* signal.

5.6 Configuration Data Block

The serial EEPROM on the MIX module provides a means with which the baseboard microcontroller may download configuration data during power up. This allows the baseboard's interconnect space to reflect specific configuration data of the MIX modules present on the stack, while allowing the microcontroller firmware to remain standard.

The hardware requirements of the module's serial EEPROM block consist of a single device, a 128 byte 93C46 serial EEPROM. This device is available in an eight pin skinny dip package which may be connected directly to the MIX interface signals. Of the 8 pins, 2 are N/C ("no-connect") pins.

The filmware portions of the serial EEPROM are discussed in Chapter 6 of this document.

5.7 Simulation

Because this module is an extremely simple design and is being prototyped in a wire wrap version, hardware simulation is considered superfluous. In this case, a trace routing or device interaction problem will not cost additional fab production cycles or considerable CAD problems.

The PAL devices however, are easily simulated using the PALASM2 software. This software includes a state simulator which allows the designer to set the trace mode for various PAL signal outputs and encode various input signal states into a line equation. By changing the input states, the designer can examine the output conditions generated. This is true for both the asynchronous and synchronous PALs designed for this module.

In the simulation for the IODCD PAL, we should investigate both valid and invalid accesses to the peripheral chips. In the PALASM source file for the I/O decode PAL (IODCD) in Appendix B, there is a section marked SIMULATION. The TRACE ON statement lists the signals which are to be examined and the SETF statements are used to change the states of the input signals.

The first SETF statement sets up an address to the PAL of 1xH, which corresponds to PPI1. The next statement specifies a word access is occurring, and finally an indication of a valid I/O cycle is given. Finally, the IOCMD signal is activated to indicate either an IORD* or IOWT* command is active. As shown in the selective trace listing, these inputs result in no chip selects since PPI1 may only be accessed as a byte wide device or in combination with PPI0.

The next group of statements changes the byte enables to a byte access which results in the /CSP1 signal being activated. The next changes the address to 0xH which is the address of PPI0. This leads to /CSP0 and /GATB0 being activated. Note the widths of the gate signals during both read and write accesses. They are

longer during writes to provide the data hold requirements of the peripheral devices. The remaining portion of the simulation sets up the byte enables to represent WORD and DWORD transfers to the PPIs. The word transfer results in the /CSP0, /CSP1, /GATB0, and /GAT32 signals being activated, while the DWORD transfer also activates the /CSP2 and /CSP3 signals. These accesses are therefore equivalent to having 16- or 32-bit PPIs.

The simulation for the I/O control PAL follows a similar process. This simulation however, allows the designer to simulate clock ticks as well as input signal state changes. The simulation is set up by placing the PAL in a reset state. The clock is cycled, then the reset is removed and an I/O read status is set. The LCLSEL*, MXCYC*, and MXCMD* signals are then set to simulate the states the PAL would normally encounter. To check the MXBS16* function, the PPI0 address is simulated on address lines /A6* through /A4*. The PAL is clocked several times to examine the operation of the MXWAIT* and MXBS16* outputs. As shown in the selective trace listing, the WAIT state machines step through their states as designed (MXWAIT* =0 correspond to MXWAIT = 1), with the MXWAIT* signal active for four clock pulses. Furthermore, the listing shows MXBS16* inactive throughout the cycle. This is desirable since 32-bit accesses are allowed for PPI0. Also we see IOCYC* go active with the MXCYC* signal and the IORD* signal go active when MXCMD* is activated. This occurs because the SWAIT signal is already active when MXCMD* is activated.

A second MIX transfer cycle is simulated to represent a full speed transfer cycle. In this cycle, we want to assure the proper WAIT machine function for command recovery and further examine the MXBS16* function. To do this, the address of PPI1 is presented to the PAL which should force MXBS16* to activate. We desire this since 32-bit cycles are only allowed for the PPI0 address offset. We also present active MXCYC* and MXCMD* signals to the PAL very soon after the end of the previous transfer cycle. This means MXCYC* has a short inactive time between cycles 1 and 2. This second simulation changes the polarity of the MXWR signal to represent a write request.

Once MXCYC* is activated we see both MXBS16* and MXWAIT* immediately activated. This ensures us we provide MXBS16* and MXWAIT* within the allotted 40 ns. In this cycle, when the command signal (MXCMD*) is activated the IOWT* signal does not come on. This occurs because the synchronous machine has not yet cycled through all of its inactive states. In the selective trace listing we see the synchronous machine continue through its inactive states even though there is a valid transfer cycle occurring. Once the machine reaches the 0001 state, IOWT* is activated. This has the effect of holding off the write command until the synchronous machine is ready to acknowledge it. Therefore, this is the command recovery mechanism built into the WAIT machine.

With acceptable PAL code simulations completed we can turn our attention to firmware requirements and module prototyping.

6.0 MODULE FIRMWARE

The MIX architecture has several firmware requirements for modules to assure proper interaction with MIX compatible baseboards and other modules. These requirements place the responsibility of supplying necessary configuration data and BIST tests on the module. While the module is responsible for supplying it, configuration data for the MIX modules is treated as an extension of the baseboard's interconnect space. This allows a MIX baseboard to use standard microcontroller code while allowing PSB agents to determine the type of MIX modules present via the baseboard's interconnect space. This structure allows dissimilar MIX modules to stack together without the need for modifying the baseboard's interconnect configuration data. Module resident BIST code allows the baseboard to test the hardware on the modules without the need for custom firmware on the baseboard. This allows different modules to stack together without the necessity of custom BIST code on the baseboard.

6.1 Serial EEPROM

The requirement placed on modules for interconnect space is the support of the National Microwire standard serial EEPROM. This is a 128 byte device which is arranged in a 64 x 16-bit configuration which allows random access of 64 word registers over four signal lines. It is programmed to contain two interconnect records which describe the configuration of the MIX module. Each interconnect register is a two byte value. The MSB contains the register protection information and the LSB contains the function's value. The baseboard's 8751 microcontroller downloads this information during power up and appends it to its own interconnect data. The *MIX Module Design Specification* describes the format which the EEPROM firmware must follow in order to function with the baseboard's 8751.

The standard format has two interconnect records, the Hardware Extension Record and the Module Specific Record as shown in Table 5. A custom module uses the standard template as a guide to defining the module's interconnect data. Required values for the registers are included in Table 5. Those registers listed as TBS in the value column have register values "To Be Specified" by the module designer.

This module defines its registers as shown in the "EX" column of Table 5. The vendor ID, 0001 (0900, 0901), represents the Intel vendor number for MULTIBUS II. Intel administers the assignment of specific vendor numbers to MULTIBUS II board manufacturers.

This Module's ID has been defined to be "TM__ MIX__MOD", with the "T" occupying register 6 and the "D" occupying register 0FH. Note that the ASCII equivalent of the string is placed in the Module ID registers.

Because this module is a wire wrap prototype, all revision level registers are given the value of 0. Finally, since this module was not designed with EPROM devices or self test capabilities, its BIST support level is defined as 0. This level means either the module hardware is not tested during power up or the baseboard firmware is responsible for supplying the module BIST code. In the case of this module, the former is true. A BIST support level of 0 implies that the Jump Table Pointer registers are assigned a value of 0.

Now that the register values are defined, the EEPROM device must simply be programmed and installed on the module. All download functions are handled by the baseboard microcontroller and do not concern the module designer.

Table 5. MIX Module Interconnect Template

Register Offset	Description	Access Local	Rights Global	Value	Ex
HARDWARE EXTENSION RECORD					
0	H/W Ext. Record Type	R/O	R/O	0914H	0914H
1	Record Length	R/O	R/O	0913H	0913H
2	H/W Ext. Type (LSB)	R/O	R/O	0910H	0910H
3	H/W Ext. Type (MSB)	R/O	R/O	0900H	0900H
4	Vendor ID, Low	R/O	R/O	TBS	0901H
5	Vendor ID, High	R/O	R/O	TBS	0900H
6-0FH	Module ID, Char 1–10	R/O	R/O	TBS	*
10H	H/W Test Rev Level	R/O	R/W1	TBS	2900H
11H	Number of Records	R/O	R/O	0901H	0901H
12H	RFU	R/O	R/O	0900H	0900H
13H	RFU	R/O	R/O	0900H	0900H
14H	RFU	R/O	R/O	0900H	0900H
MODULE SPECIFIC RECORD					
15H	Mod. Specific Record Type	R/O	R/O	09F0H	09F0H
16H	Record Length	R/O	R/O	TBS	0909H
17H	EPROM Entry Table Ptr., MSB	R/O	R/O	TBS	0900H
18H	EPROM Entry Table Ptr., Byte 3	R/O	R/O	TBS	0900H
19H	EPROM Entry Table Ptr., Byte 2	R/O	R/O	TBS	0900H
1AH	EPROM Entry Table Ptr., LSB	R/O	R/O	TBS	0900H
1BH	RFU	R/O	R/O	0900H	0900H
1CH	H/W Rev Level	R/O	R/W1	TBS	2900H
1DH	BIST Test Support Level	R/O	R/W1	TBS	2900H
1EH	RFU	R/O	R/O	0900H	0900H
1EF	RFU	R/O	R/O	0900H	0900H
20H	RFU	R/O	R/O	0900H	0900H
21H–3FH	Module Specific Info	Module Defined			

The 93C64 device can be programmed using a standard EEPROM programmer which supports the device. Because the device is electrically erasable, it is possible for the EEPROM to be programmed by the baseboard's microcontroller. At the time of this module design, this support was not included in the baseboard 8751 firmware.

6.2 BIST Support Levels

The MIX architecture provides several BIST support levels as described in the *Mix Module Design Specification*. Inclusion of EPROMS and BIST firmware on modules allows power up verification of the module hardware problems are encountered.

This module does not incorporate EPROM or BIST support in its design. This is defined as BIST support level 0. Although level 0 allows the stipulation that the MIX baseboard carry the module BIST code in its EPROM, this module design assumes no power up testing is performed on the module hardware. This option was chosen because of the simplicity of the module hardware used in the design. Furthermore, addition of module EPROMs would have further complicated the I/O control block portion of the design, which would have lessened the simple example intentions of this module design.

7.0 MODULE PROTOTYPING

Once the module is fully defined and its schematic has been completed, a final check of the modules electrical characteristics can be made. The main concern here is the power requirements of the module. We should ensure the devices used in this design meet the power consumption requirements listed for the MIX interface. Once we have assured these characteristics, layout and actual wire wrapping and debug can be done.

7.1 Electrical Considerations

Before the actual module assembly begins, we must check the module design against the electrical specifications for the MIX interface. The two major issues here involve the drive characteristics of the devices which connect to the MIX signals and the total maximum power consumption of the module. Both of these parameters may affect how the module operates with the MIX baseboard or other modules.

The drive characteristic check deals with the type of driver used on the MIX bus. The MIX specification states that no module may use open collector devices to drive the MIX bus signals. A quick check of the module device specifications assure us that the module meets this requirement.

The second area of concern is the power consumption of the module. The *MIX Module Design Specification* states that each module may consume up to 3 Amps of +5 VDC. Again, by going through each device's spec sheet we can compute the maximum and typical power consumption of the module.

In reviewing the components used for this module, we find the maximum current draw on the +5V power is just under 2.5 Amps for the entire module, including the LEDs. Although we find this simple module is surprisingly near the 3 Amp limit, a typical module design would not implement 32 LEDs which draw about 25 mA each.

7.2 Wire Wrap Layout

Now that the electrical requirements of the MIX module are verified, actual prototyping will be done. The layout portion of the prototyping requires knowledge of the signal paths and component interconnection. The schematic capture program used in this design contains a program which generates a netlist from the schematic output. With this netlist and the MIX MOD2 Breadboard Module, we are ready to begin the layout.

In laying out components for the MIX design, careful attention should be given to the trace length requirements given in the *MIX Module Design Specification*. Although this is merely a wire wrap version of the final product, adhering to the trace length rules may save some debugging headaches later. The Breadboard module offers the standard MIX module width and MIX interface footprint, however, the board length is much longer than a standard module. This added length is provided to allow an engineer to prototype surface mount designs using DIP components which require more design area.

The module layout follows the signal paths between the MIX interface and I/O devices. The termination resistors and data transceivers are placed nearest the MIX connector. The two PAL devices and EEPROM also reside adjacent to the connector due to their direct connection to interface signals. The PPI devices and 82510 are placed next, and the output drivers, LEDs, and connectors are placed at the edges of the board. This layout structure provides a natural signal propagation from the MIX interface out to the edges of the module and ensures the trace lengths are kept to a minimum. Once the components are placed, they may be connected via wire wrap.

7.3 Module Debug

Once the module has been completely wired up, it is ready for verification and debug. Although most designers are confident to immediately initiate a "smoke

test", it is wise to verify the device connections with the netlist provided by the schematic capture package. This is time consuming but is easily done using a simple ohmmeter. Once this has been checked out, the module may be installed on the baseboard for final debug.

The final validation for this module was performed on a MIX 386/020A baseboard with a MIX MOD1 Test Module in module slot 0 and the wire wrap module in slot 1. In this stacking configuration, the wire wrap module's I/O address space ranges from 0800H to 0C00H. This means the I/O address offsets encoded in the IODCD PAL are added to the base address of 0800H to obtain the actual addresses for the module's devices. For example, the baseboard 386 microprocessor would access PPI0 at 098xH.

To facilitate ease of baseboard I/O access, firmware was placed on the MIX baseboard which contained a DIN III monitor configured (with the proper 82510 port address) to run with MIX MOD1 in module slot 0. This provides a serial interface for the baseboard and removes the need for additional system boards (except the CSM/001) for the module debug.

In debugging the board, we must check out several things, access to the boards serial EEPROM, parallel port functionality, and 82510 functionality. The serial EEPROM interface is verified by reading the baseboards interconnect space. DIN III provides an input interconnect (ii) function which aids in this task. The *MIX 386/020A Baseboard User's Guide* shows the first H/W Extension Record should begin at interconnect offset 65H. This register can be checked for a value of 14H. If the 14H is found, the module ID registers (6BH–74H) can be read. If the module's ID is present (in ASCII), the serial interface is functioning.

Parallel ports may be verified by using the LEDs designed on the module's PPI port C or by using a logic analyzer. Using the LEDs requires setting up port C of the PPI as a simple output port and then making output accesses to port C to toggle the LEDs. To use a logic analyzer, direct accesses to the PPIs may be checked, or the port outputs may be checked. To verify the port A connector, ports A and B must be configured as an output port since bits 0 and 1 of port B control the direction and enable pin of the port A drivers.

The addressing scheme used in this module places the port A data bus at x0H, the port B data bus at x4H, port C's data at x8H, and the PPI's control port at xCH. For the PPI verification, all PPI ports are set up as simple output channels. This is done by writing 80H to ports 098CH, 099CH, 09ACH, and 09BCH. As shown on sheet 10 of the schematic, port B bits 0 and 1 affect the function of the port A driver chip. Therefore, a 3 is output to 0984H to enable the port A drivers and set their direction as outputs. Data is then output to both ports A and C to verify the module operation.

The 82510 device may be similarly checked out. With the module in slot 1, the 82510 internal registers begin at I/O address 09C0H. The chip can be verified by setting its 35 internal registers to configure a loopback mode to verify data transmit and receive data consistency.

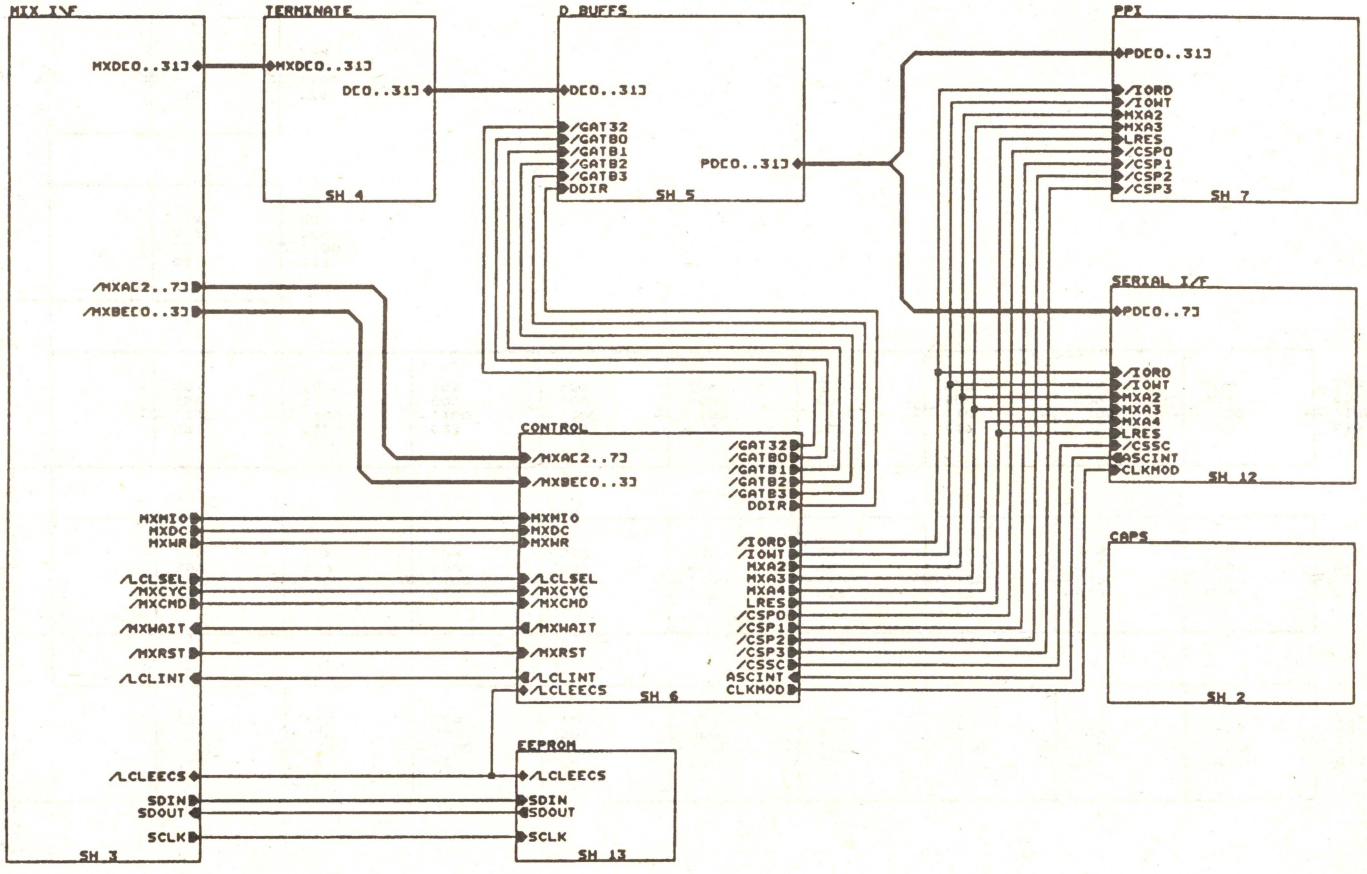
Once the 82510, PPIs, and EEPROM operation have been verified, a high level of design confidence is achieved, and production cycles may proceed as needed.

8.0 CONCLUSIONS

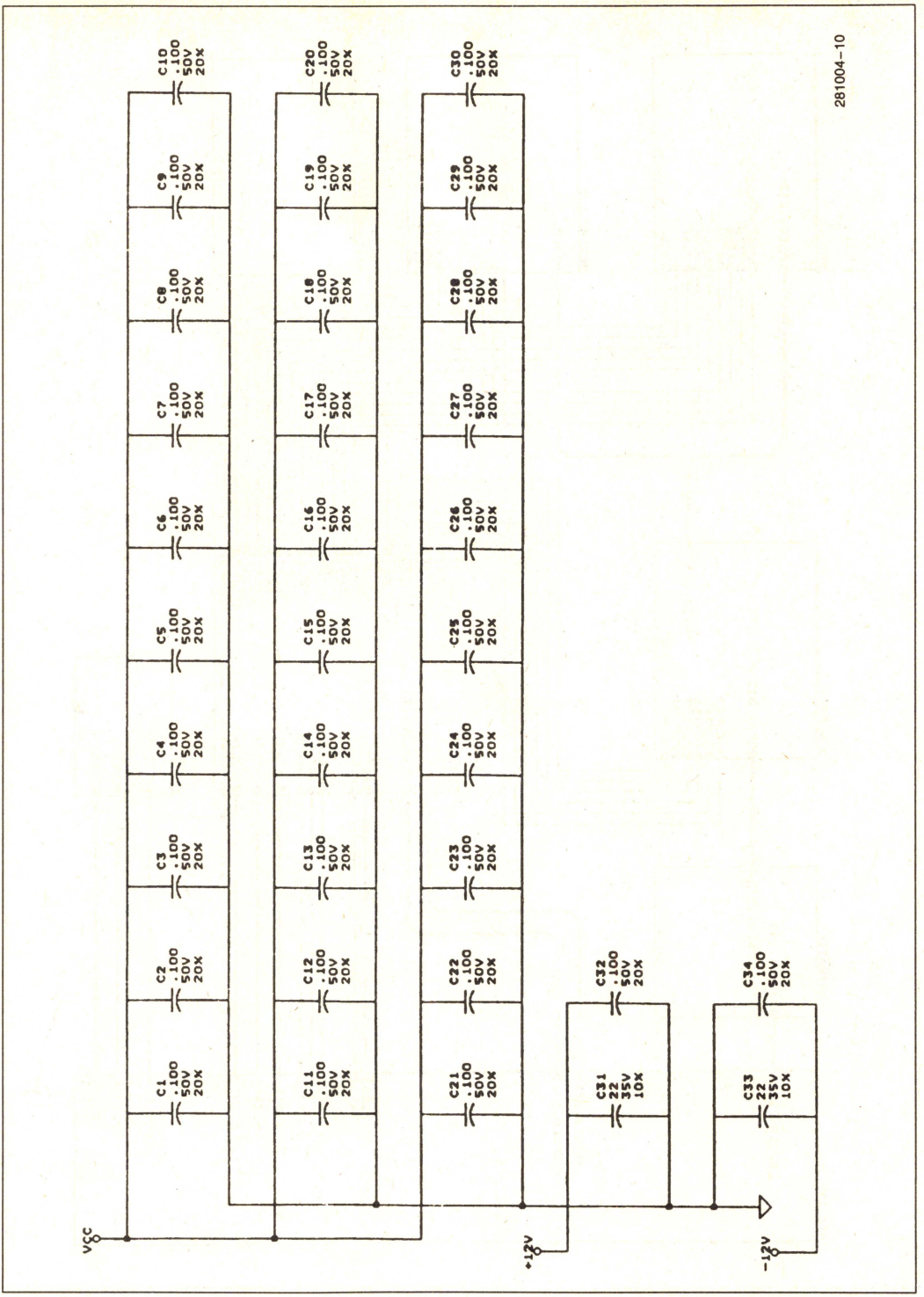
Although this module design is a simple one and has been done simply for display and example purposes, it conveys the considerations required in a MIX module design effort. Furthermore, the control and data drivers blocks of the module may be used as a design guide in many simple slave I/O module designs.

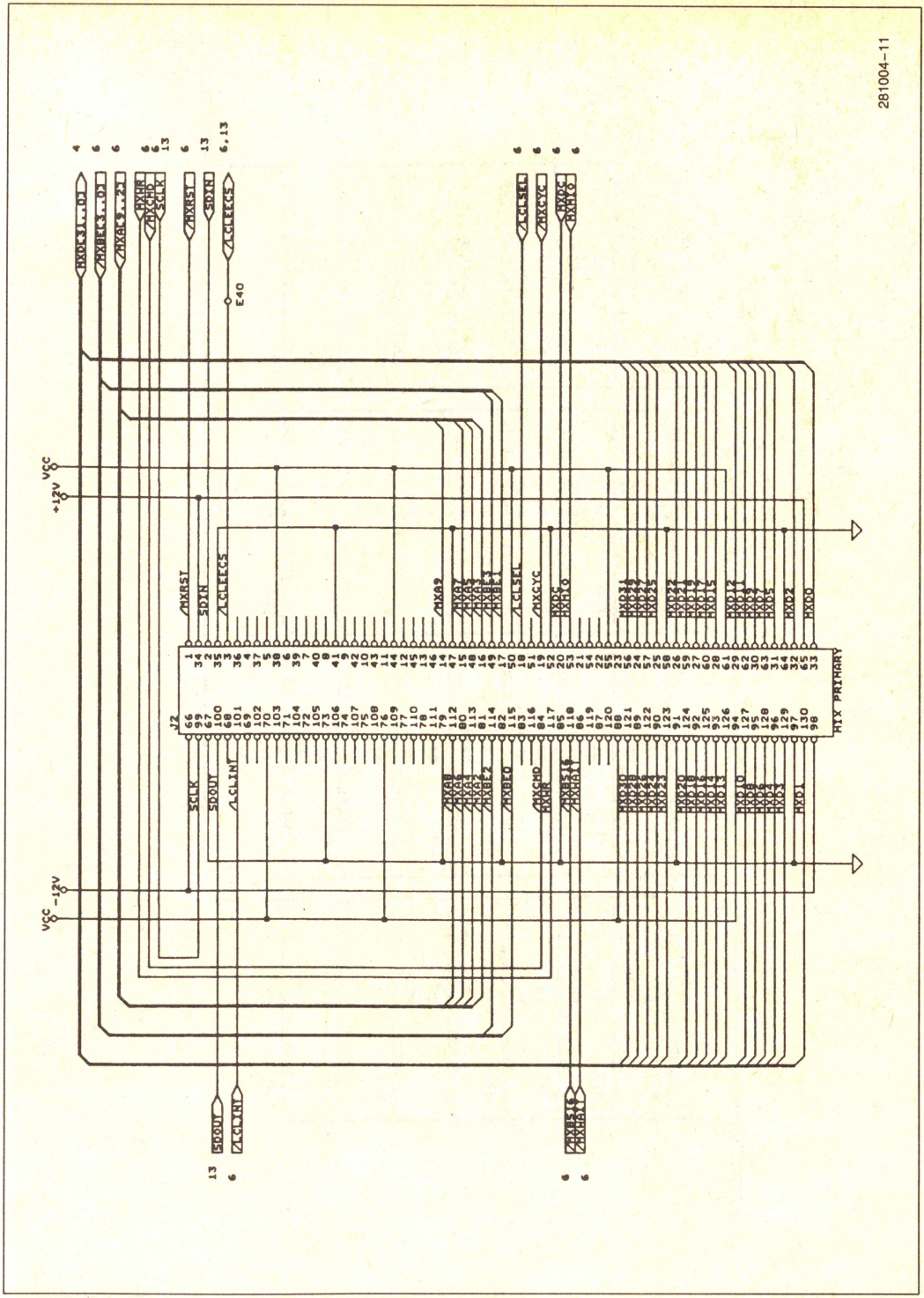
The module schematics and PAL source codes have been included in Appendices A and B as an aid to designing simple I/O modules and as a basis for more complex module designs.

APPENDIX A
MIX MODULE SCHEMATICS

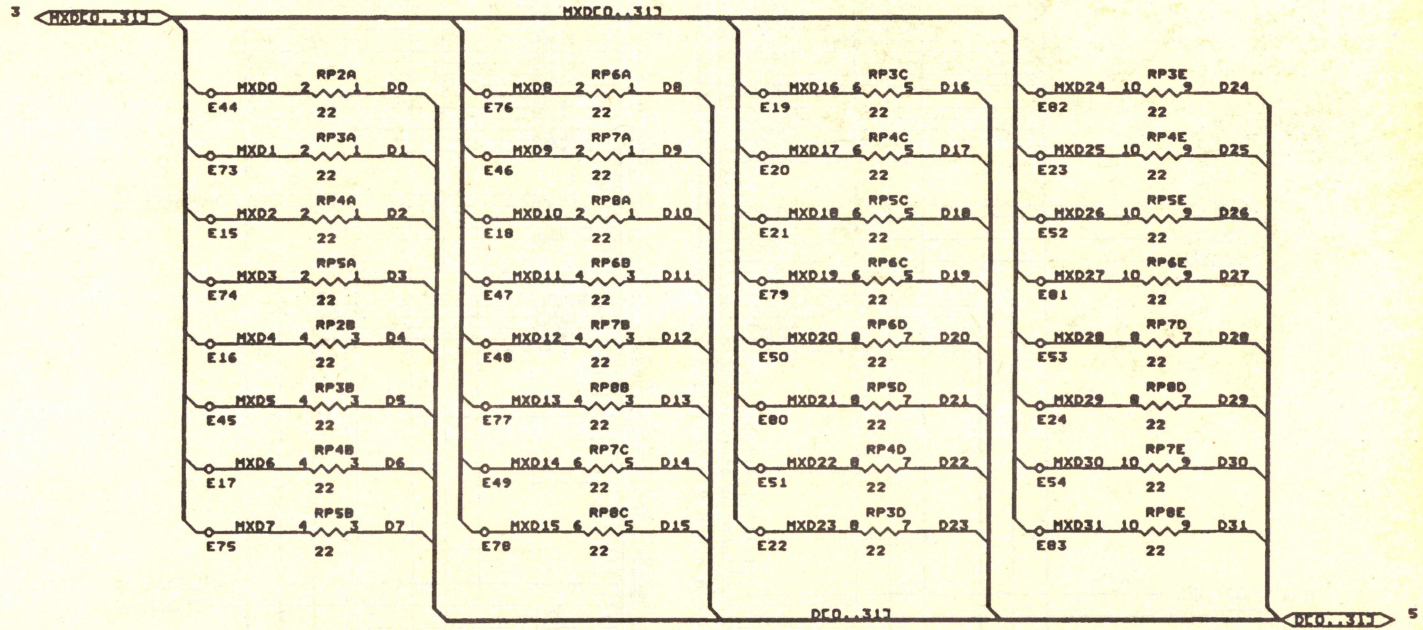

MIX I\F
MXDCO..31J
/MXAC2..7J
/MXBECO..3J
MXMIO
MXDC
MXWR
/LCLSEL
/MXCYC
/MXCMD
/MXWAIT
/MXRST
/LCLINT
/LCLEECS
SDIN
SDOUT
SCLK
SH 3
TERMINATE
MXDCO..31J
DCO..31J
SH 4
D BUFFS
DCO..31J
/GAT32
/GATBO
/GATB1
/GATB2
/GATB3
DDIR
PDCO..31J
SH 5
CONTROL
/MXAC2..7J
/MXBECO..3J
MXMIO
MXDC
MXWR
/LCLSEL
/MXCYC
/MXCMD
/MXWAIT
/MXRST
/LCLINT
/LCLEECS
/GAT32
/GATBO
/GATB1
/GATB2
/GATB3
DDIR
/IORD
/IOWT
MXA2
MXA3
MXA4
LRES
/CSPO
/CSP1
/CSP2
/CSP3
/CSSC
ASCINT
CLKMOD
SH 6
EEPROM
/LCLEECS
SDIN
SDOUT
SCLK
SH 13
PPI
PDCO..31J
/IORD
/IOWT
MXA2
MXA3
LRES
/CSPO
/CSP1
/CSP2
/CSP3
SH 7
SERIAL I/F
PDCO..7J
/IORD
/IOWT
MXA2
MXA3
MXA4
LRES
/CSSC
ASCINT
CLKMOD
SH 12
CAPS
SH 2

281004-9

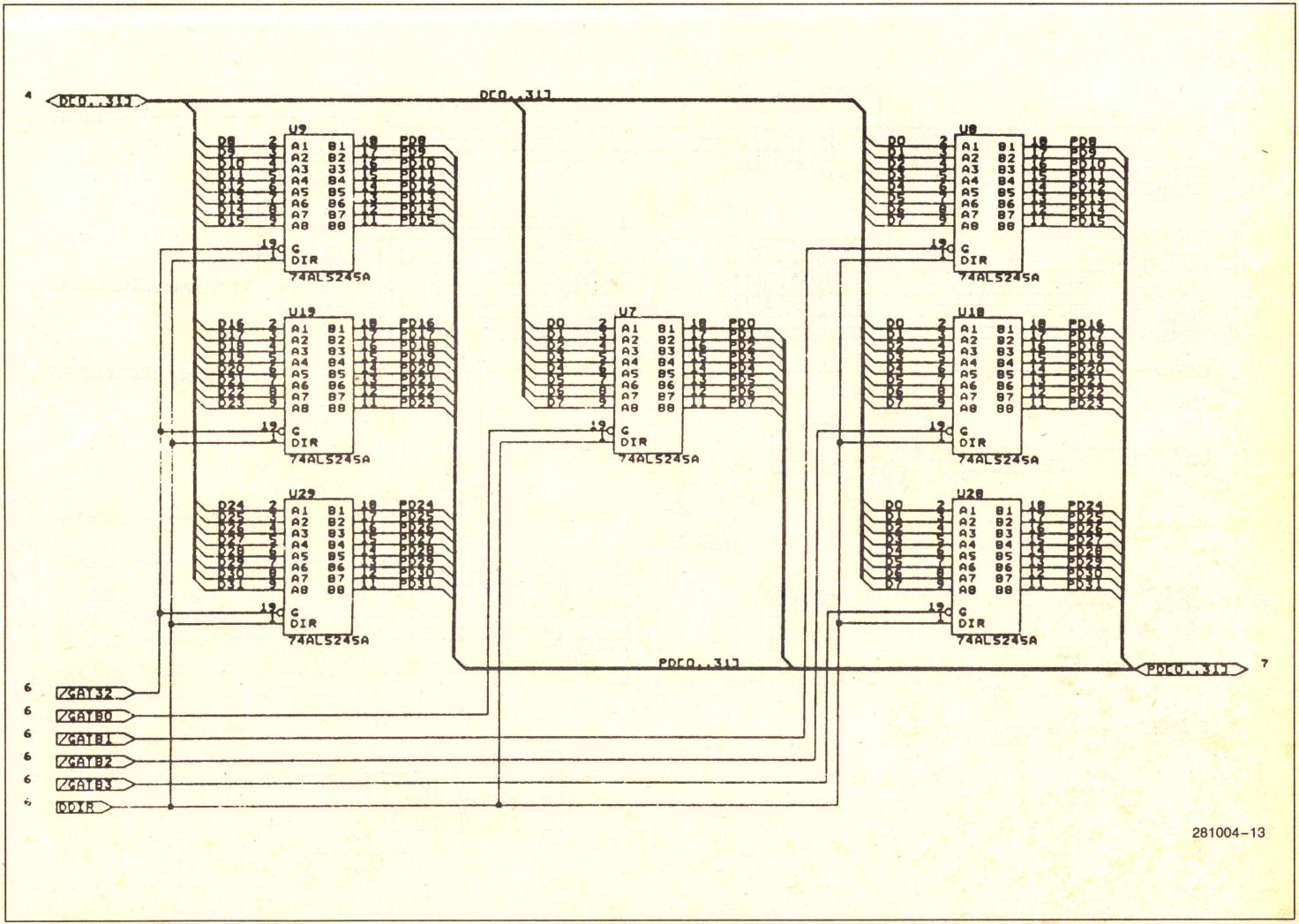

C1 .100 50V 20%
C2 .100 50V 20%
C3 .100 50V 20%
C4 .100 50V 20%
C5 .100 50V 20%
C6 .100 50V 20%
C7 .100 50V 20%
C8 .100 50V 20%
C9 .100 50V 20%
C10 .100 50V 20%
C11 .100 50V 20%
C12 .100 50V 20%
C13 .100 50V 20%
C14 .100 50V 20%
C15 .100 50V 20%
C16 .100 50V 20%
C17 .100 50V 20%
C18 .100 50V 20%
C19 .100 50V 20%
C20 .100 50V 20%
C21 .100 50V 20%
C22 .100 50V 20%
C23 .100 50V 20%
C24 .100 50V 20%
C25 .100 50V 20%
C26 .100 50V 20%
C27 .100 50V 20%
C28 .100 50V 20%
C29 .100 50V 20%
C30 .100 50V 20%
C31 22 35V 10%
C32 .100 50V 20%
C33 22 35V 10%
C34 .100 50V 20%
VCC
+12V
-12V
281004-10


281004-11
HXD[31..0]
/HXBE[3..0]
/HXA[31..2]
HXCMD
HXACK
SCLK
/HXRST
SDIN
/LCLKECS
E40
/LCLSFL
/HXCYC
HXRDY
HXINTO
VCC
+12V
VCC
-12V
J2
HIX PRIMARY
SDOUT
/LCLINT
/HXBE3
/HXBE2
/HXBE1
/HXBE0
/HXA31
/HXA30
/HXA29
/HXA28
/HXA27
/HXA26
/HXA25
/HXA24
/HXA23
/HXA22
/HXA21
/HXA20
/HXA19
/HXA18
/HXA17
/HXA16
/HXA15
/HXA14
/HXA13
/HXA12
/HXA11
/HXA10
/HXA9
/HXA8
/HXA7
/HXA6
/HXA5
/HXA4
/HXA3
/HXA2
HXD31
HXD30
HXD29
HXD28
HXD27
HXD26
HXD25
HXD24
HXD23
HXD22
HXD21
HXD20
HXD19
HXD18
HXD17
HXD16
HXD15
HXD14
HXD13
HXD12
HXD11
HXD10
HXD9
HXD8
HXD7
HXD6
HXD5
HXD4
HXD3
HXD2
HXD1
HXD0

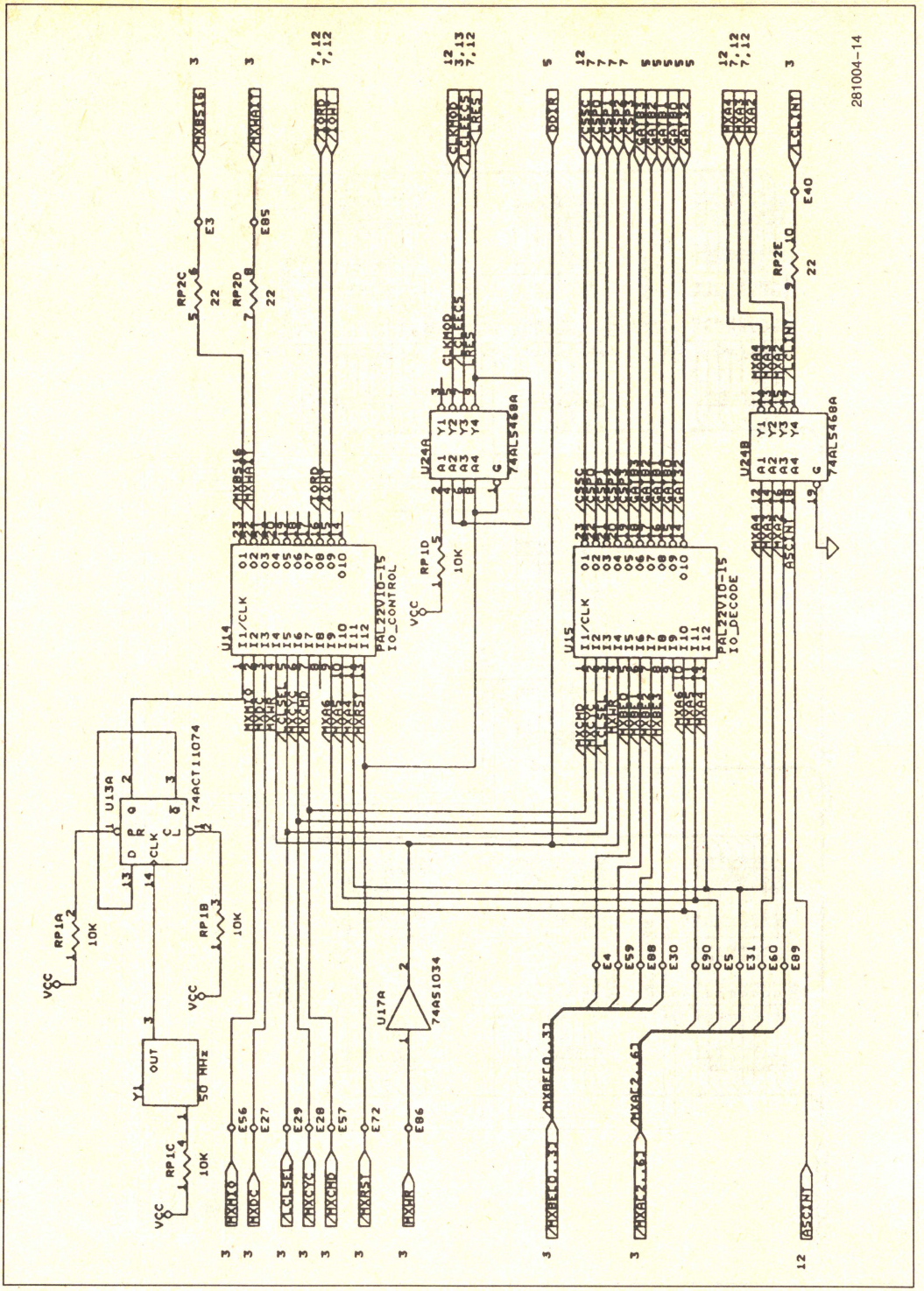
5

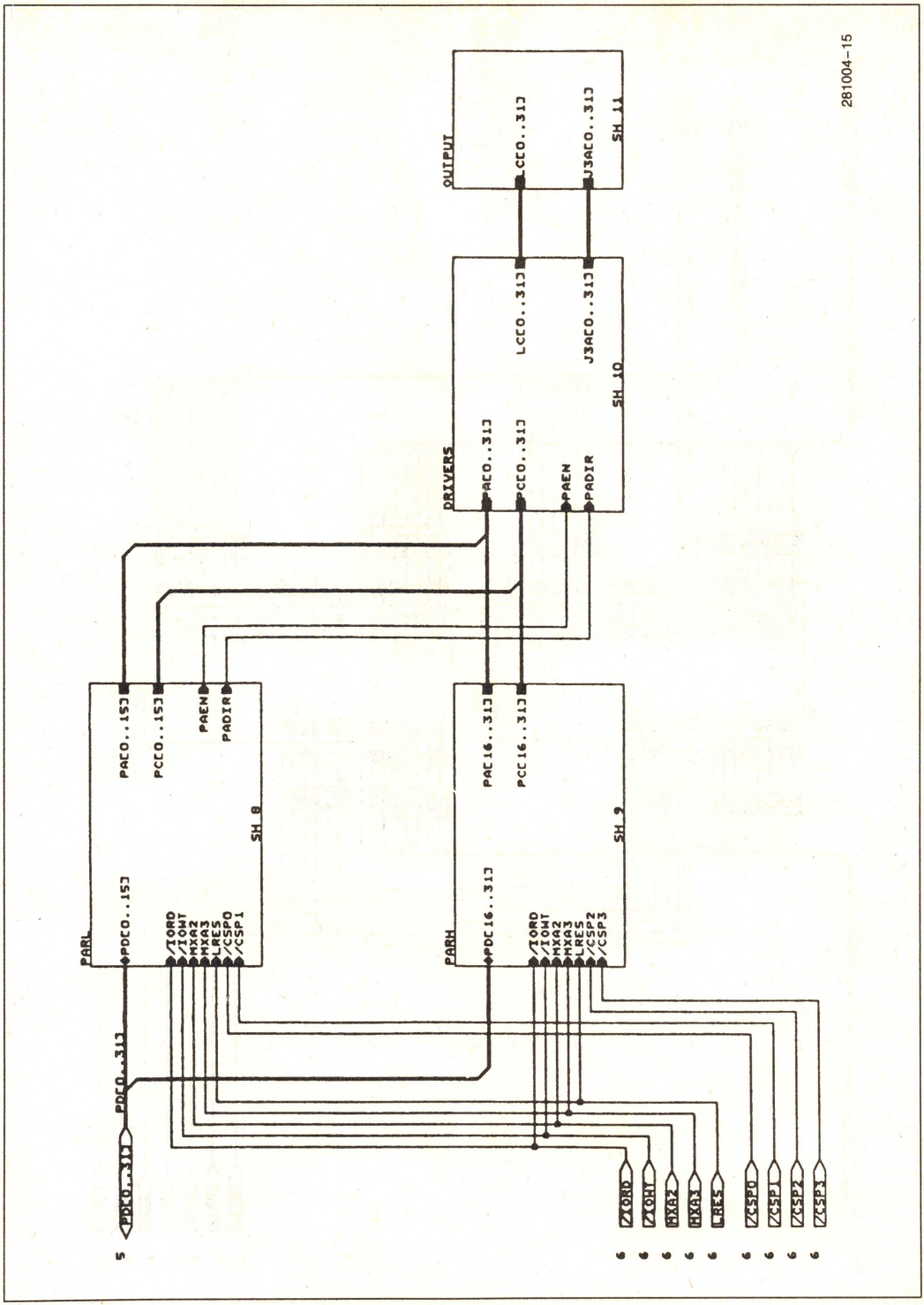

3
MXDF0..31]
MXDF0..31]
RP2A
MXD0 2 1 D0
E44
22
RP3A
MXD1 2 1 D1
E73
22
RP4A
MXD2 2 1 D2
E15
22
RP5A
MXD3 2 1 D3
E74
22
RP2B
MXD4 4 3 D4
E16
22
RP3B
MXD5 4 3 D5
E45
22
RP4B
MXD6 4 3 D6
E17
22
RP5B
MXD7 4 3 D7
E75
22
RP6A
MXD8 2 1 D8
E76
22
RP7A
MXD9 2 1 D9
E46
22
RP8A
MXD10 2 1 D10
E18
22
RP6B
MXD11 4 3 D11
E47
22
RP7B
MXD12 4 3 D12
E48
22
RP8B
MXD13 4 3 D13
E77
22
RP7C
MXD14 6 5 D14
E49
22
RP8C
MXD15 6 5 D15
E78
22
RP3C
MXD16 6 5 D16
E19
22
RP4C
MXD17 6 5 D17
E20
22
RP5C
MXD18 6 5 D18
E21
22
RP6C
MXD19 6 5 D19
E79
22
RP6D
MXD20 8 7 D20
E50
22
RP5D
MXD21 8 7 D21
E80
22
RP4D
MXD22 8 7 D22
E51
22
RP3D
MXD23 8 7 D23
E22
22
RP3E
MXD24 10 9 D24
E82
22
RP4E
MXD25 10 9 D25
E23
22
RP5E
MXD26 10 9 D26
E52
22
RP6E
MXD27 10 9 D27
E81
22
RP7D
MXD28 8 7 D28
E53
22
RP8D
MXD29 8 7 D29
E24
22
RP7E
MXD30 10 9 D30
E54
22
RP8E
MXD31 10 9 D31
E83
22
DF0..31]
DF0..31]
5

281004-12

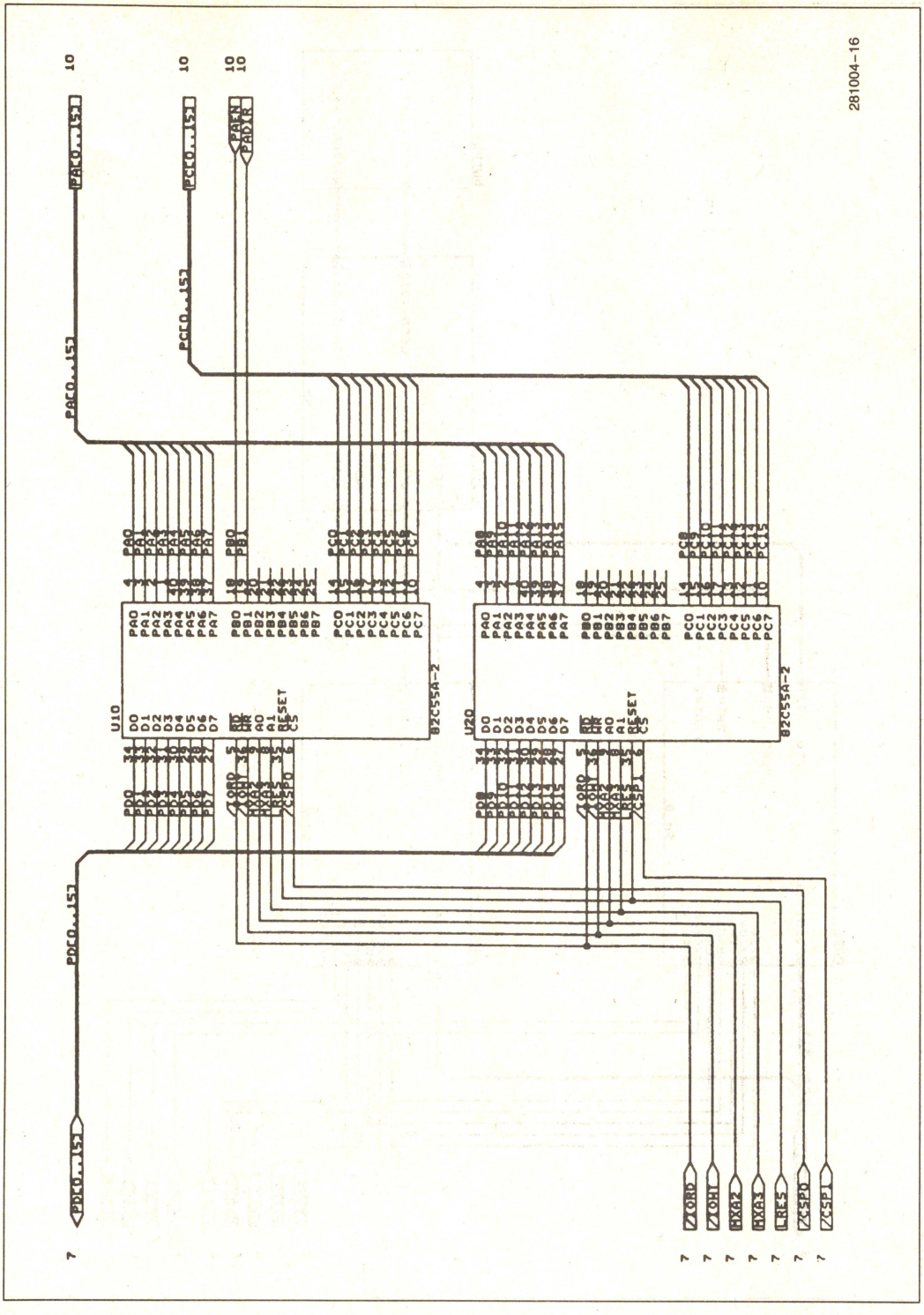

PD[0..31]
U9
U19
U29
U7
U8
U18
U28
74ALS245A
/GAT32
/GAT80
/GAT81
/GAT82
/GAT83
DDIR

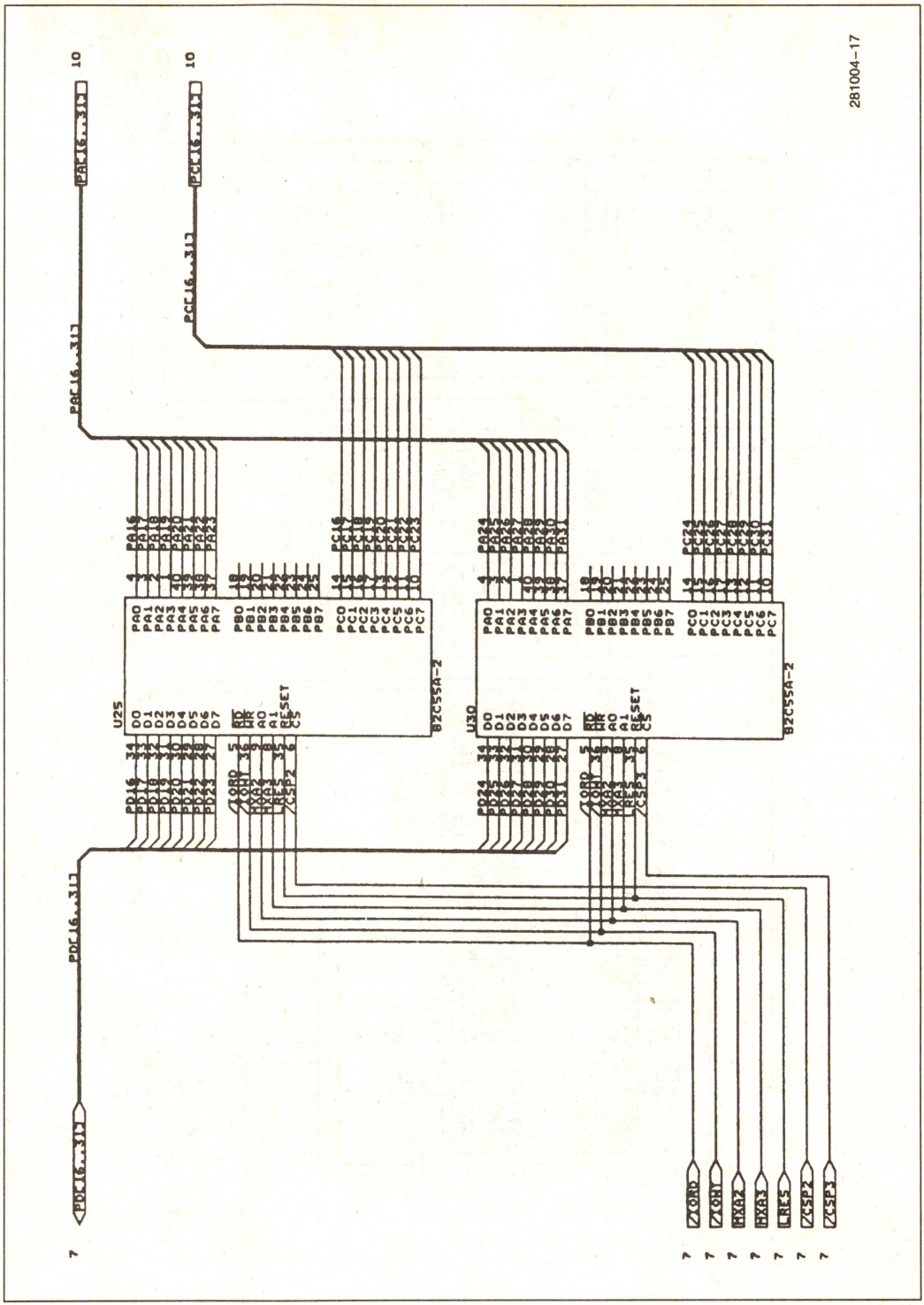
281004-13

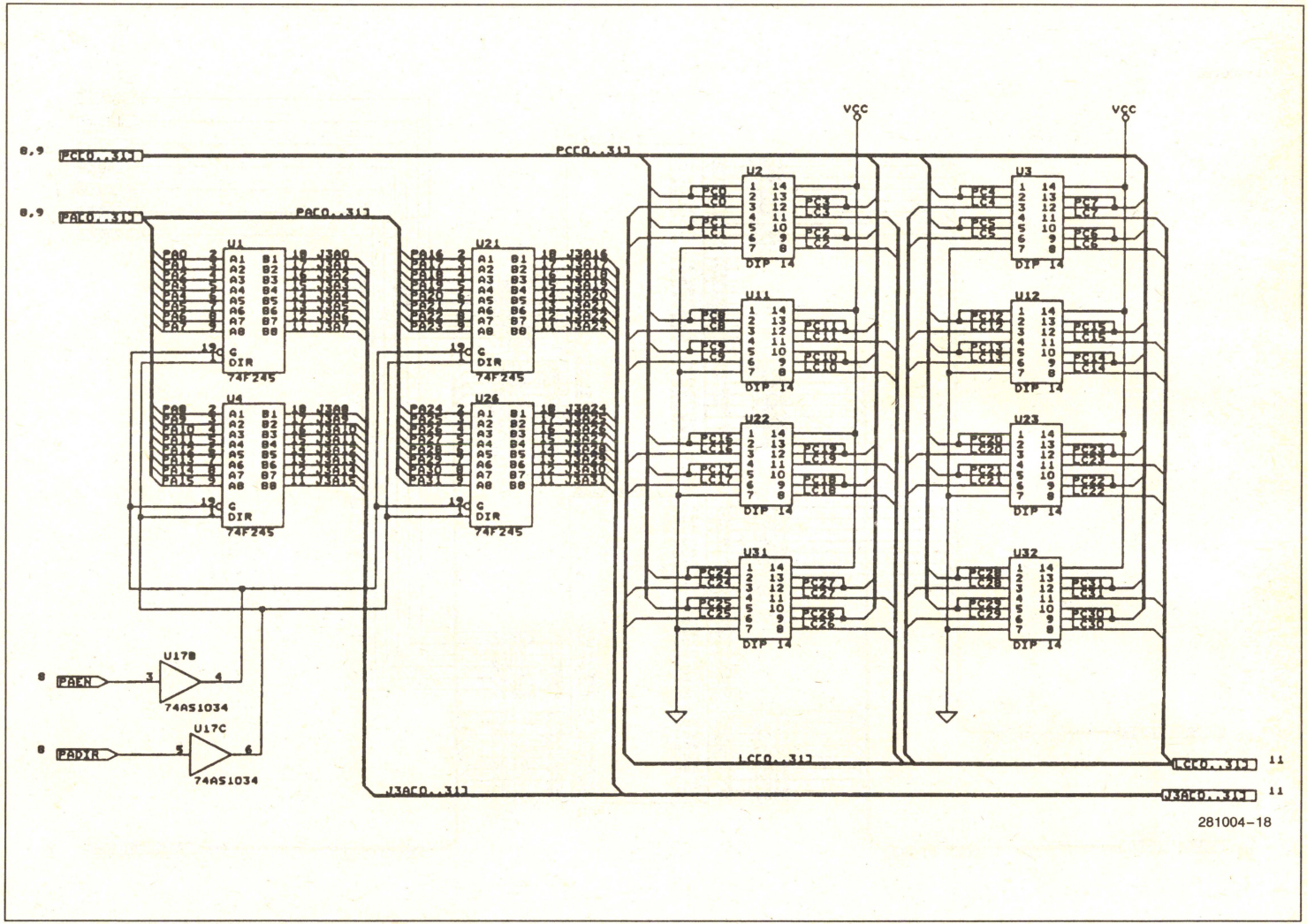

281004-14
U14
PAL22V10-15
IO_CONTROL
U15
PAL22V10-15
IO_DECODE
U13A
74ACT11074
U17A
74AS1034
U24A
U24B
74ALS468A
Y1
OUT
50 MHz
RP1A
RP1B
RP1C
RP1D
RP2C
RP2D
RP2E
10K
22
VCC
/CLKINT
/CLSEL
/HXRST
HXHR
HXDIR
/HXCMD
/HXCYC
/HXBE0...3
/HXBE2...6
/HXAB2...6
/HXAB7...12
/HXBS16
/HXBA1
/LOBP
/LOBW
/CLKOUT
/CLKOUT2
/RES
/CSSC
/CSSP0
/CSSP1
/CSSP2
/CSSP3
/CSSP4
/CSSP5
/CSSP6
/CSSP7
/CSSP8
/CSSP9
/CSSP10
/CSSP11
/CSSP12
/CSSP13
/CSSP14
/CSSP15
/CSSP16
/CSSP17
/CSSP18
/CSSP19
/CSSP20
/CSSP21
/CSSP22
/CSSP23
/CSSP24
/CSSP25
/CSSP26
/CSSP27
/CSSP28
/CSSP29
/CSSP30
/CSSP31
/CSSP32
/CSSP33
/CSSP34
/CSSP35
/CSSP36
/CSSP37
/CSSP38
/CSSP39

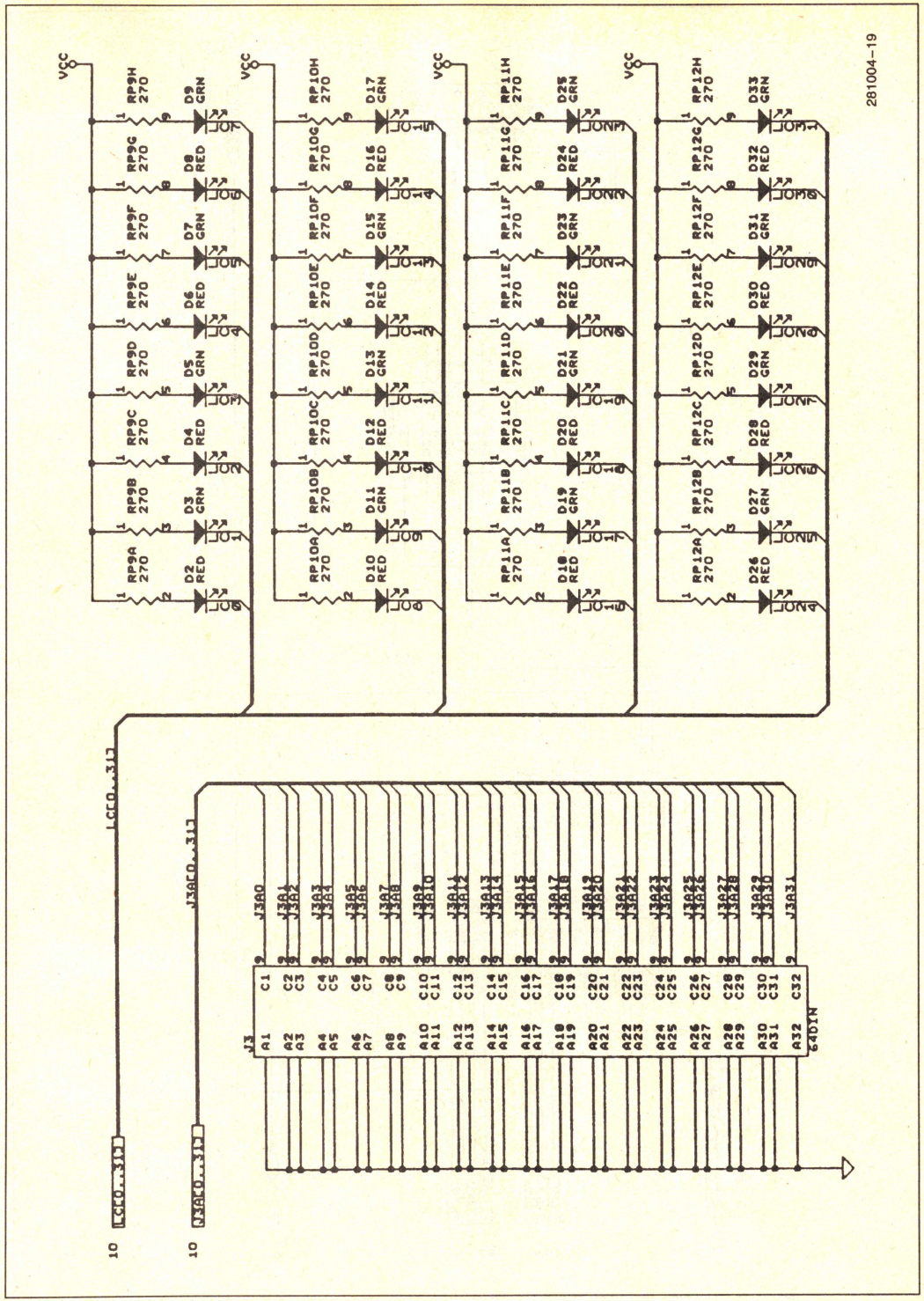

281004-15
OUTPUT
LCC0..31J
J3AC0..31J
SH_11
DRIVERS
PAC0..31J
PCC0..31J
PAEN
PADIR
SH_10
PAC0..15J
PCC0..15J
PAEN
PADIR
SH_8
PARL
PDC0..15J
/IORD
/IOWT
MXA2
MXA3
LRES
/CSP0
/CSP1
PAC16..31J
PCC16..31J
SH_9
PARH
PDC16..31J
/IORD
/IOWT
MXA2
MXA3
LRES
/CSP2
/CSP3
PDC0..31J
5
6
ZIORD
ZIOWT
MXA2
MXA3
LRES
ZCSP0
ZCSP1
ZCSP2
ZCSP3

5

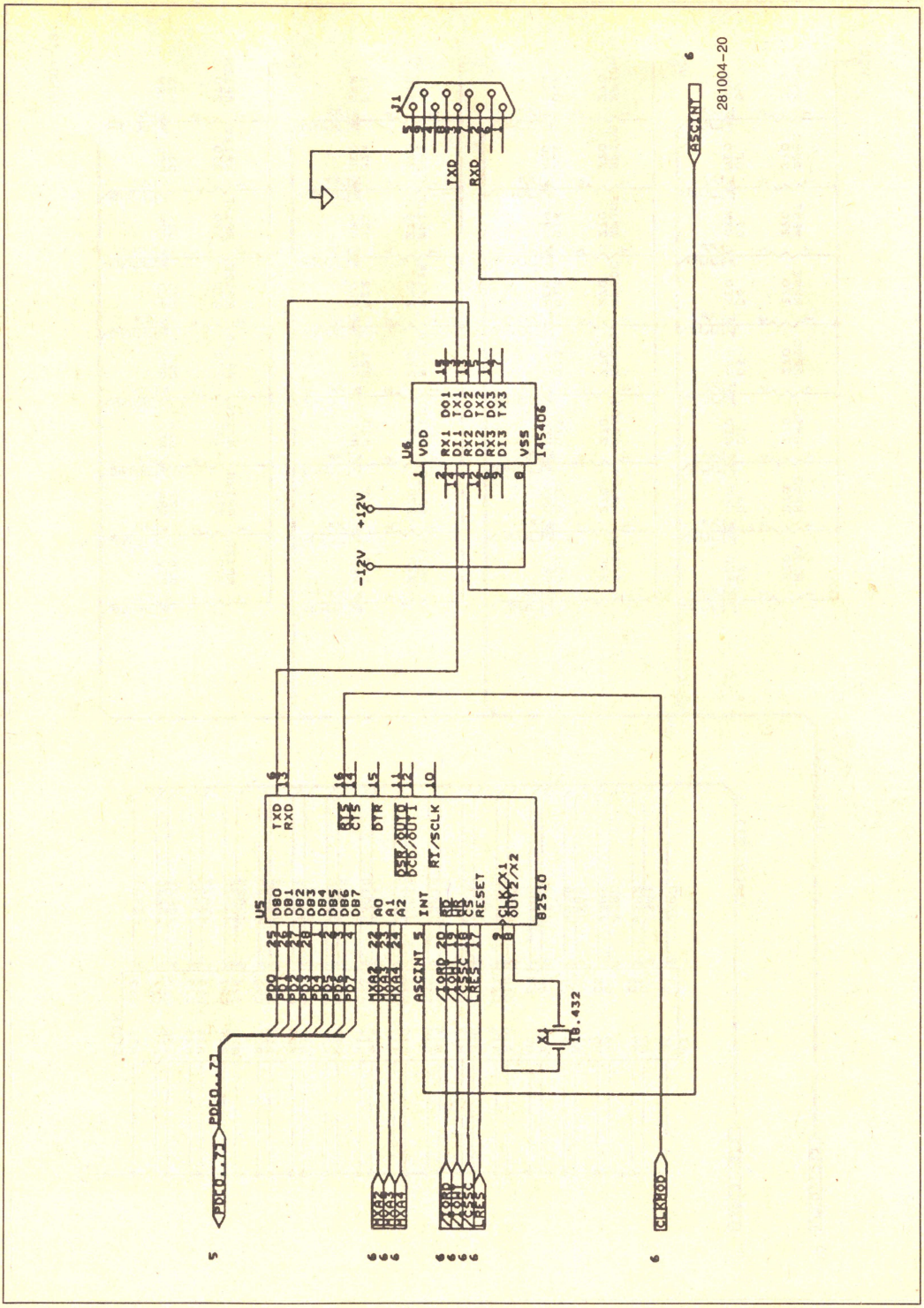

281004-16
U10
U20
82C55A-2
PA0...15
PB0...15
PC0...15
PA0
PB0
PC0
PA8
PB8
PC8
RESET
ZCSPD
ZCSPJ
ZXAX
ZXHI
ZXORD

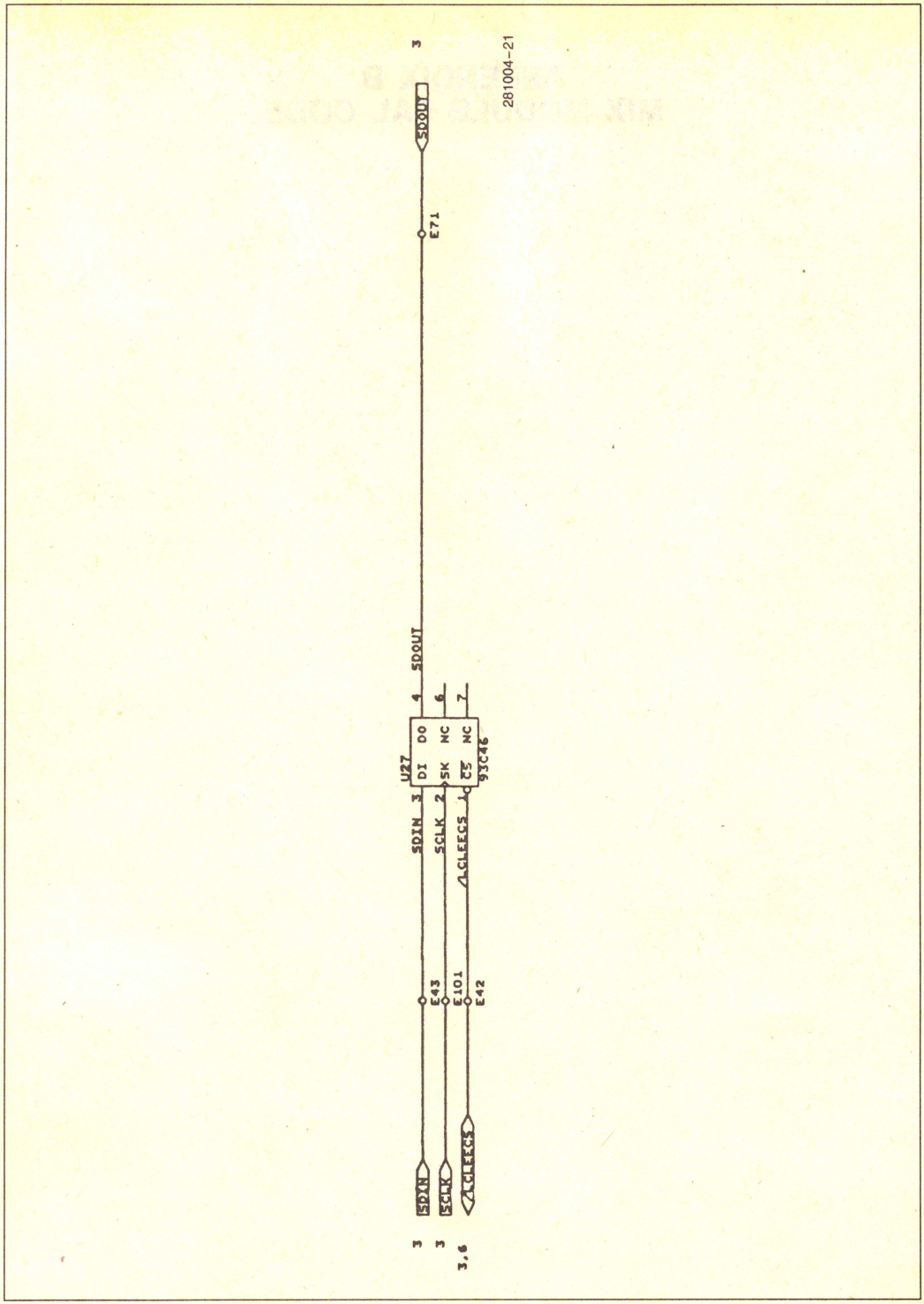

281004-17
PAC16..31
PCC16..31
PBC16..31
PDC16..31
U25
U30
82C55A-2
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
D0
D1
D2
D3
D4
D5
D6
D7
RD
WR
A0
A1
RESET
CS
ZCWR
ZCWR
HXA2
HXA3
CRES
ZCSP2
ZCSP3


8,9
PCC0..31]
PCC0..31]
8,9
PAC0..31]
PAC0..31]
VCC
VCC
U1
U21
U2
U3
U4
U26
U11
U12
U22
U23
U31
U32
74F245
DIP 14
U17B
U17C
74AS1034
PAEN
PADIR
LCC0..31]
J3AC0..31]
11
281004-18


281004-19
VCC
RP9H 270
RP9G 270
RP9F 270
RP9E 270
RP9D 270
RP9C 270
RP9B 270
RP9A 270
D9 GRN
D8 RED
D7 GRN
D6 RED
D5 GRN
D4 RED
D3 GRN
D2 RED
RP10H 270
RP10G 270
RP10F 270
RP10E 270
RP10D 270
RP10C 270
RP10B 270
RP10A 270
D17 GRN
D16 RED
D15 GRN
D14 RED
D13 GRN
D12 RED
D11 GRN
D10 RED
RP11H 270
RP11G 270
RP11F 270
RP11E 270
RP11D 270
RP11C 270
RP11B 270
RP11A 270
D25 GRN
D24 RED
D23 GRN
D22 RED
D21 GRN
D20 RED
D19 GRN
D18 RED
RP12H 270
RP12G 270
RP12F 270
RP12E 270
RP12D 270
RP12C 270
RP12B 270
RP12A 270
D33 GRN
D32 RED
D31 GRN
D30 RED
D29 GRN
D28 RED
D27 GRN
D26 RED
LGC0..31]
J3AC0..31]
10
J3
64DIN

5

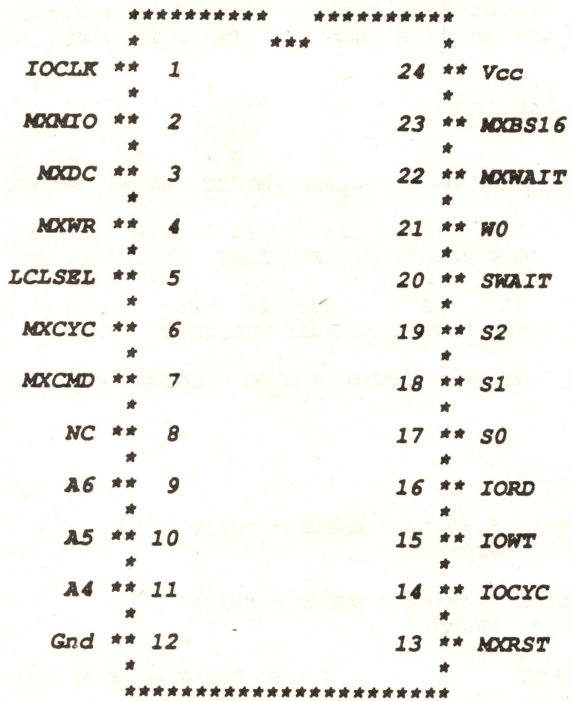

281004-20
J1
TXD
RXD
ASCINT
U6
VDD
RX1 DO1
RI1 TX1
RI2 TX2
DI2 DO2
RX2 DO3
DI3 TX3
VSS
145406
+12V
-12V
U5
TXD
RXD
DTR
RT/SCLK
DB0
DB1
DB2
DB3
DB4
DB5
DB6
DB7
A0
A1
A2
INT
RESET
CLK/X1
82510
ASCINT
PD0..7
PDC0..7
X1
18.432
CLKMOD


281004-21
3
SDOUT
E71
SDOUT
4
6
7
U27
DI DO
SK NC
CS NC
93C46
SDIN 3
SCLK 2
ACLEECS 1
E43
E101
E42
3
3
3,6
SDIN
SCLK
ACLEECS

APPENDIX B
MIX MODULE PAL CODE

Title : IOCTL PAL
Pattern : U14
Rev : 1.02
Author : Erik A. Steeb
Company : INTel Corporation
Date : 10/2/89

Monolithic Memories PAL22V10-15



281004-22

5

```
; created by eas
; (C) - COPYRIGHT INTeL Corporation, 1989
 TITLE IOCTL PAL
 PATTERN U14
 REVISION 1.02
 AUTHOR ERIK A STEEB
 COMPANY INTeL Corporation
 DATE 10/2/89

;Revision 1.02 corrects a problem in the synchronous wait machine
;which brought the machine back to the 0000 state when IOCYC went
;inactive.

CHIP IO_CONTROL PAL22V10

;PINS  1      2      3      4       5       6       7     8   9
     IOCLK MXMIO  MXDC MXWR /LCLSEL /MXCYC /MXCMD  NC /A6

;PINS 10     11     12     13      14      15      16    17  18
     /A5    /A4    GND /MXRST /IOCYC /IOWT  /IORD  S0  S1

;PINS 19     20     21     22      23    24    25
      S2  SWAIT    W0 /MXWAIT /MXBS16 VCC INIT

STRING IO_CYCLE '/MXRST * /MXMIO * MXDC * LCLSEL * MXCYC'

EQUATIONS

IOCYC    =  IO_CYCLE

IORD     =  IO_CYCLE * /MXWR * MXCMD * SWAIT
         +  IORD  *  MXCMD

IOWT     =  IO_CYCLE *  MXWR * MXCMD * SWAIT
         +  IOWT  *  MXCMD

MXBS16.TRST = IOCYC                    ; IO_CYCLE + 15 ns Tri-State

MXBS16   =  IO_CYCLE *  A6             ; BS16* asserted any access
         +  IO_CYCLE *  A5             ; except to PPI 0.
         +  IO_CYCLE *  A4
```

281004-24

```
;  *** SYNCHRONOUS PORTION OF WAIT MACHINE ***

INIT.RSTF = MXRST                    ; Global reset of register outputs

S0      :=   S0 *  S1
         +   S0 * /S2
         +   S0 *  SWAIT
         +   S1 * /S2 * SWAIT

S1      :=  /MXCMD *  S0 * /S2 * /SWAIT
         +   S0 *  S1 * /S2
         +  /S0 *  S1 *  S2
         +  /S0 *  S2 *  SWAIT
         +   S1 * /S2 *  SWAIT

S2      :=   S0 *  S1
         +   S0 *  S2 * /SWAIT
         +  /S0 * /S1 *  S2 *  SWAIT
         +  /S0 * /S1 *  SWAIT *  MXCMD

SWAIT   :=  /S0 *  SWAIT
         +   S0 *  S1 *  SWAIT
         +   S0 * /S1 *  S2 *  SWAIT
         +   IO_CYCLE *  MXWAIT * /S0 * /S1 * /S2 * /SWAIT

;  *** ASYNCHRONOUS PORTION OF WAIT MACHINE ***

W0      =   IO_CYCLE * /SWAIT * W0
         +  IO_CYCLE *  SWAIT * MXWAIT

MXWAIT.TRST = IOCYC          ;( IO_CYCLE + 15 ns TRI-STATE )

MXWAIT  =   IO_CYCLE *  SWAIT * MXWAIT
         +  IO_CYCLE * /W0

;
;  DESCRIPTION
;
;  The IOCYC variable is used by the tri-state outputs for MXBS16* and

;  MXWAIT* to facilitate the "logical false before tri-state" rule.
;
```

281004-25

5

```
SIMULATION

TRACE_ON   /IORD /IOWT /IOCYC /MXCYC /MXCMD S0 S1 S2 SWAIT W0 /MXWAIT
/MXBS16
SETF /IOCLK MXRST /MXCMD /LCLSEL /MXCYC
CLOCKF
CLOCKF
SETF /MXRST /MXMIO MXDC /MXWR     ; set I/O read status
SETF /A6 /A5 /A4                  ; set PPI0 address
CLOCKF
CLOCKF
SETF LCLSEL                       ; select the module
CLOCKF
CLOCKF
SETF MXCYC                        ; start a valid cycle
CLOCKF                            ; provide plenty of clocks
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
SETF MXCMD                        ; issue the command
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
SETF /MXCMD                       ; release the command
CLOCKF
SETF /MXCYC                       ; end transfer cycle
SETF /MXRST /MXMIO MXDC MXWR      ; set I/O write status
SETF /A6 /A5 A4                   ; set PPI1 address
CLOCKF
SETF MXCYC                        ; begin transfer cycle
CLOCKF
SETF MXCMD                        ; issue command
```

281004-26

```
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
SETF /MXCMD                    ; release command
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
SETF /MXCYC                    ; end cycle
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
CLOCKF
SETF /LCLSEL                   ; deselect module
CLOCKF
CLOCKF
TRACE_OFF
```

281004-27

```
PALASM SIMULATION, V2.23 - MARKET RELEASE (2-1-88)
(C) - COPYRIGHT MONOLITHIC MEMORIES INC, 1988
PALASM SIMULATION SELECTIVE TRACE LISTING

Title     : IOCTL PAL                    Author    : ERIK A STEEB
Pattern   : U14                          Company   : INTeL Corporation
Revision  : 1.02                         Date      : 10/2/89

PAL22V10
Page :  1
         g c cgg  c   cg  c  c g  c  c  c   c  c  c
/IORD    HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH
/IOWT    HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH
/IOCYC   HHHHHHHHHH HHHHHHHHHH LLLLLLLLLL LLLLLLLLLL
/MXCYC   HHHHHHHHHH HHHHHHHHHH LLLLLLLLLL LLLLLLLLLL
/MXCMD   HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH
 S0      LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL
 S1      LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL
 S2      LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL
 SWAIT   LLLLLLLLLL LLLLLLLLLL LLHHHHHHHH HHHHHHHHHH
 W0      LLLLLLLLLL LLLLLLLLLL LLHHHHHHHH HHHHHHHHHH
/MXWAIT  HHHHHHHHHH HHHHHHHHHH LLLLLLLLLL LLLLLLLLLL
/MXBS16  HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH

Page :  2
          c  c  c   c  c  c  c g  c  c  c   c  c  c
/IORD    HHHHHHHHHH HHHHHHHHHH LLLLLLLLLL LLLLLLLLLL
/IOWT    HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH
/IOCYC   LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL
/MXCYC   LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL
/MXCMD   HHHHHHHHHH HHHHHHHHHH LLLLLLLLLL LLLLLLLLLL
 S0      LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LHHHHHHHHH
 S1      LLLLLLLLLL LLLLLLLLLL LLLLLHHHHH HHHHHHHLLL
 S2      LLLLLLLLLL LLLLLLLLLL LLHHHHHHLL LLLLHHHHHH
 SWAIT   HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH
 W0      HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH
/MXWAIT  LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL
/MXBS16  HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH

Page :  3
          c  c  c   c  cg  cg  gg  cg   c g  c  c  c
/IORD    LLLLLLLLLL LLLLHHHHHH HHHHHHHHHH HHHHHHHHHH
/IOWT    HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HHHHHHHHLL
/IOCYC   LLLLLLLLLL LLLLLLLLHH HHHHHLLLLL LLLLLLLLLL
/MXCYC   LLLLLLLLLL LLLLLLLLHH HHHHHLLLLL LLLLLLLLLL
/MXCMD   LLLLLLLLLL LLLLHHHHHH HHHHHHHHHH LLLLLLLLLL
 S0      HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HHLLLLLLLL
 S1      LLLLLLLLLL LLLLLLHHHH HHHHHHHHLL LLLLLLLLLL
 S2      LLLLLLLLLL LLLLLLLLLL LLHHHHHHHH HHHHHLLLLL
 SWAIT   HHHLLLLLLL LLLLLLLLLL LLLLLLLLLL LLLLLLLLHH
 W0      HHHHHHHHHH HHHHHHHHLL LLLLLLLLLL LLLLLLLLHH
/MXWAIT  LLLHHHHHHH HHHHHHHHHZ ZZZZZZLLLL LLLLLLLLLL
/MXBS16  HHHHHHHHHH HHHHHHHHHZ ZZZZZZLLLL LLLLLLLLLL
```

281004-28

```
Page :  4
          c  c  c   c  c  c   c  c  c  c   c  c  c
/IORD    HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH
/IOWT    LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL
/IOCYC   LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL
/MXCYC   LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL
/MXCMD   LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL
 S0      LLLLLLLLLL HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH
 S1      LLLLHHHHHH HHHHHHLLLL LLLLLLLLLL LLLLLLLLLL
 S2      LHHHHHHLLL LLLHHHHHHL LLLLLLLLLL LLLLLLLLLL
 SWAIT   HHHHHHHHHH HHHHHHHHHH HHLLLLLLLL LLLLLLLLLL
 W0      HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH
/MXWAIT  LLLLLLLLLL LLLLLLLLLL LLHHHHHHHH HHHHHHHHHH
/MXBS16  LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL

Page :  5
          c  c  c   c  c  cg  c  c  c  c   c  c  cg
/IORD    HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH
/IOWT    LLLLLLLLLL LLLLLLLHHH HHHHHHHHHH HHHHHHHHHH
/IOCYC   LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LLLLLLLLLH
/MXCYC   LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LLLLLLLLLH
/MXCMD   LLLLLLLLLL LLLLLLLHHH HHHHHHHHHH HHHHHHHHHH
 S0      HHHHHHHHHH HHHHHHHHHH HHHHHHHHLL LLLLLLLLLL
 S1      LLLLLLLLLL LLLLLLLLLH HHHHHLLLLL LLLLLLLLLL
 S2      LLLLLLLLLL LLLLLLLLLL LLHHHHHHHH HLLLLLLLLL
 SWAIT   LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL
 W0      HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HHHHHHHHHL
/MXWAIT  HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH
/MXBS16  LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LLLLLLLLLH

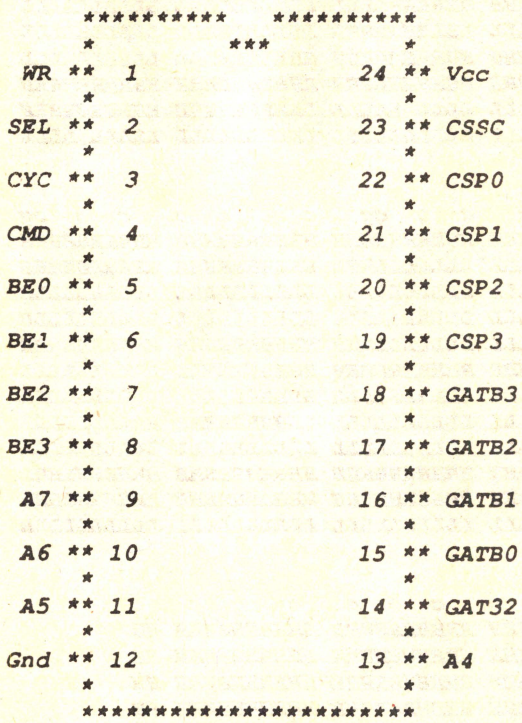
Page :  6
          c  c  c   c  c  c   c  cg  c   c
/IORD    HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HH
/IOWT    HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HH
/IOCYC   HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HH
/MXCYC   HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HH
/MXCMD   HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HH
 S0      LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LL
 S1      LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LL
 S2      LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LL
 SWAIT   LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LL
 W0      LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LL
/MXWAIT  ZZZZZZZZZZ ZZZZZZZZZZ ZZZZZZZZZZ ZZ
/MXBS16  ZZZZZZZZZZ ZZZZZZZZZZ ZZZZZZZZZZ ZZ
```

281004-29

5

Title : IODCD PAL
Pattern : U15
Rev : 1.01
Author : Erik A. Steeb
Company : INTel Corporation
Date : 5/24/89

Monolithic Memories PAL22V10-15



281004-23

```
; created by eas
; (C) - COPYRIGHT INTeL Corporation, 1989
 TITLE IODCD PAL
 PATTERN U15
 REVISION 1.01
 AUTHOR ERIK STEEB
 COMPANY INTeL Corporation
 DATE 5/24/89

CHIP IO_DECODE PAL22V10

;PINS   1    2    3    4    5    6    7    8    9   10  11
       WR /SEL /CYC /CMD /BE0 /BE1 /BE2 /BE3  NC /A6 /A5

;PINS  12    13   14     15     16     17
       GND  /A4 /GAT32 /GATB0 /GATB1 /GATB2

;PINS   18     19    20    21    22    23  24   25
      /GATB3 /CSP3 /CSP2 /CSP1 /CSP0 /CSSC VCC INIT

STRING  CYCLE    'SEL * CYC'
STRING  GATE_RD  'SEL * CYC * /WR * CMD'
STRING  GATE_WT  'SEL * CYC *  WR'

EQUATIONS

GATB0  = GATE_RD * /A6 * /A5 * /A4
       + GATE_WT * /A6 * /A5 * /A4
       + GATE_RD *  A6 * /A5
       + GATE_WT *  A6 * /A5

GATB1  = GATE_RD * /A6 * /A5 *  A4 * /BE1 *  BE0
       + GATE_WT * /A6 * /A5 *  A4 * /BE1 *  BE0

GATB2  = GATE_RD * /A6 *  A5 * /A4 * /BE1 *  BE0
       + GATE_WT * /A6 *  A5 * /A4 * /BE1 *  BE0

GATB3  = GATE_RD * /A6 *  A5 *  A4 * /BE1 *  BE0
       + GATE_WT * /A6 *  A5 *  A4 * /BE1 *  BE0

GAT32  = GATE_RD * /A6 * /A5 * /A4 *  BE3 *  BE2 *  BE1 * BE0
       + GATE_WT * /A6 * /A5 * /A4 *  BE3 *  BE2 *  BE1 * BE0
       + GATE_RD * /A6 * /A5 * /A4 * /BE3 * /BE2 *  BE1 * BE0
       + GATE_WT * /A6 * /A5 * /A4 * /BE3 * /BE2 *  BE1 * BE0
```

281004-30

5

```
CSP0    = CYCLE   * /A6 * /A5 * /A4 *  BE3 *  BE2 *  BE1 * BE0
        + CYCLE   * /A6 * /A5 * /A4 * /BE3 * /BE2 *  BE1 * BE0
        + CYCLE   * /A6 * /A5 * /A4 * /BE3 * /BE2 * /BE1 * BE0

CSP1    = CYCLE   * /A6 * /A5 * /A4 *  BE3 *  BE2 *  BE1 * BE0
        + CYCLE   * /A6 * /A5 * /A4 * /BE3 * /BE2 *  BE1 * BE0
        + CYCLE   * /A6 * /A5 *  A4 * /BE3 * /BE2 * /BE1 * BE0

CSP2    = CYCLE   * /A6 * /A5 * /A4 *  BE3 *  BE2 *  BE1 * BE0
        + CYCLE   * /A6 *  A5 * /A4 * /BE3 * /BE2 * /BE1 * BE0

CSP3    = CYCLE   * /A6 * /A5 * /A4 *  BE3 *  BE2 *  BE1 * BE0
        + CYCLE   * /A6 *  A5 *  A4 * /BE3 * /BE2 * /BE1 * BE0

CSSC    = CYCLE   *  A6 * /A5 * /BE3 * /BE2 * /BE1 * BE0


 ; DESCRIPTION;:
;
; NAME CHANGES       SCHEMATIC  to  PIN LIST
;
;                      /LCLSEL         /SEL
;                      /MXCYC          /CYC
;                      /MXCMD          /CMD
;                       MXWR            WR
;                      /MXBEx          /BEx
;                      /MXAx           /Ax
;
;  Read and write accesses distinquished in "GAT" equations to meet
;  82C55A-2 data hold times for writes and MIX spec data off times
;  for reads.
;
; I/O DECODE SCHEME AS FOLLOWS:
;
;  Byte  Parallel Port @ offset xxx000xxxx binary (first = 0xH)
;  Byte  Parallel Port @ offset xxx001xxxx binary (first = 1xH)
;  Byte  Parallel Port @ offset xxx010xxxx binary (first = 2xH)
;  Byte  Parallel Port @ offset xxx011xxxx binary (first = 3xH)
;  Word  Parallel Port @ offset xxx000xxxx (Combo of two byte ports)
;  DWord Parallel Port @ offset xxx000xxxx (Combo of four byte ports)
;  Serial Port         @ offset xxx10xxxxx  (first = 4xH & 5xH)
;
;  NOTE: Address lines /MXA9, /MXA8, and /MXA7 not used so decode
;        is mirrored every 128 bytes.
```

281004-31

```
SIMULATION

TRACE_ON  /SEL /CYC /CMD WR /CSP0 /CSP1 /CSP2 /CSP3 /CSSC
          /GATB0 /GATB1 /GATB2 /GATB3 /GAT32
SETF      SEL
SETF      /WR /A6 /A5 A4                ; Set PPI1 address read
SETF      /BE3 /BE2 BE1 BE0             ; Word access
SETF      CYC                           ; Begin valid cycle
SETF      CMD
SETF      /CMD
SETF      /CYC                          ; end cycle
SETF      /BE1                          ; Byte access
SETF      CYC                           ; Begin cycle
SETF      CMD
SETF      /CMD
SETF      /CYC                          ; end cycle
SETF      WR                            ; Write access
SETF      CYC                           ; Begin cycle
SETF      CMD
SETF      /CMD
SETF      /CYC                          ; end cycle
SETF      /A4                           ; Set PPI0 address
SETF      CYC                           ; Begin cycle
SETF      CMD
SETF      /CMD
SETF      /CYC                          ; end cycle
SETF      BE1                           ; Word access
SETF      CYC                           ; Begin valid cycle
SETF      CMD
SETF      /CMD
SETF      /CYC                          ; end cycle
SETF      /WR BE3 BE2                   ; Dword access read
SETF      CYC                           ; Begin valid cycle
SETF      CMD
SETF      /CMD
SETF      /CYC                          ; end cycle
SETF      A6 A5                         ; Set invalid address read
SETF      CYC                           ; Begin valid cycle
SETF      CMD
SETF      /CMD
SETF      /CYC                          ; end cycle
SETF      /BE3 BE2 BE1 /BE0             ; Set invalid byte enable
SETF      CYC                           ; Begin valid cycle
SETF      CMD
SETF      /CMD
SETF      /CYC                          ; end cycle
TRACE_OFF
```

281004-32

5

```
PALASM SIMULATION, V2.23 - MARKET RELEASE (2-1-88)
(C) - COPYRIGHT MONOLITHIC MEMORIES INC, 1988
PALASM SIMULATION SELECTIVE TRACE LISTING

Title     : IODCD PAL                    Author    : ERIK STEEB
Pattern   : U15                          Company   : INTeL Corporation
Revision  : 1.01                         Date      : 5/24/89

PAL22V10
        gggggggggg gggggggggg gggggggggg gggggggggg
/SEL    LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL LLLLLLLLLL
/CYC    XXXLLLHHLL LHHLLLHHLL LHHLLLHHLL LHHLLLHHLL
/CMD    XXXXLHHHHL HHHHLHHHHL HHHHLHHHHL HHHHLHHHHL
 WR     XLLLLLLLLL LLHHHHHHHH HHHHHHHLLL LLLLLLLLLL
/CSP0   XHHHHHHHHH HHHHHHHHLL LHHLLLHHLL LHHHHHHHHH
/CSP1   XXHHHHHHLL LHHLLLHHHH HHHLLLHHLL LHHHHHHHHH
/CSP2   XHHHHHHHHH HHHHHHHHHH HHHHHHHHLL LHHHHHHHHH
/CSP3   XHHHHHHHHH HHHHHHHHHH HHHHHHHHLL LHHHHHHHHH
/CSSC   XHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH
/GATB0  XHHHHHHHHH HHHHHHHHLL LHHLLLHHHL HHHHHHHHHH
/GATB1  XXHHHHHHHL HHHLLLHHHH HHHHHHHHHH HHHHHHHHHH
/GATB2  XHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH
/GATB3  XHHHHHHHHH HHHHHHHHHH HHHHHHHHHH HHHHHHHHHH
/GAT32  XHHHHHHHHH HHHHHHHHHH HHHLLLHHHL HHHHHHHHHH

Page :  2
        gg
/SEL    LL
/CYC    LH
/CMD    HH
 WR     LL
/CSP0   HH
/CSP1   HH
/CSP2   HH
/CSP3   HH
/CSSC   HH
/GATB0  HH
/GATB1  HH
/GATB2  HH
/GATB3  HH
/GAT32  HH
```

281004-33

intel®

AP-437

APPLICATION NOTE

Designing a MULTIBUS II CPU Board with the Message Passing Coprocessor (MPC)—82389

5

JOE BECKER
TECHNICAL MARKETING ENGINEER

TIM MOSTAD
TECHNICAL MARKETING MANAGER

August 1993

Order Number: 281022-001

Designing a MULTIBUS II CPU Board with the Message Passing Coprocessor (MPC)—82389

CONTENTS

PAGE

Chapter 1: INTRODUCTION 5-223

1.0 WHO SHOULD READ THIS DOCUMENT? 5-223

1.1 Overview 5-223

1.2 Objectives 5-223

Chapter 2: PSB INTERFACING 5-224

2.0 INTERFACING TO THE IEEE/ANSI 1296 PARALLEL SYSTEM BUS 5-224

2.1 IEEE/ANSI 1296 Fundamentals Which Must Be Understood to Design a Board 5-224

2.2 PSB Connections to the MPC and Local Logic 5-226

2.2.1 Providing PSB Address for Read/Write Cycles on PSB 5-228

2.2.2 Reset and Power-Down Signals, Bus Clock, and Slot Arbitration Status Latching 5-228

Chapter 3: INTERCONNECT SPACE IMPLEMENTATION CIRCUITRY 5-231

3.0 THE CONCEPT OF INTERCONNECT SPACE 5-231

3.1.1 Interface between the MPC and 8751 Microcontroller 5-231

3.1.2 Controlling Board Resource Configurations through Interconnect Space 5-233

3.1.3 Reset Operations 5-233

CONTENTS

PAGE

Chapter 4: Intel Microprocessor Interfacing Examples 5-234

4.1 80186 Interface Example 5-234

4.2 80286 Interface Example 5-236

4.3 80386 Interface Example 5-238

Chapter 5: DMA CONTROLLER INTERFACE 5-240

5.0 DMA CONTROLLER INTERFACE 5-240

5.1 MPC-ADMA Interface Signals .. 5-242

5.2 Transfer Mode 5-242

5.3 Duty Cycle 5-242

5.4 Alternate Design Method 5-242

Chapter 6: DUAL-PORT MEMORY INTERFACE 5-244

6.0 DUAL-PORT MEMORY 5-244

6.1 Interfacing Signals 5-244

6.2 Dual-Port Memory Operation ... 5-246

Chapter 7: SUGGESTED READING ... 5-246

CHAPTER 1: INTRODUCTION

1.0 WHO SHOULD READ THIS DOCUMENT?

Readers of this document may be considering the design of an intelligent MULTIBUS II board. Concise descriptions of the various possibilities of a board's design are presented in this document, in particular, a focus on the use of the MPC (Message Passing Coprocessor or 82389), which provides a majority of the interface logic. Descriptions here represent summaries of other documentation available, and experience with the design of Intel's MULTIBUS II product line.

After reading this document, one will feel confident that the design of a MULTIBUS II CPU board is no more complex than the design of another bus system. Actually, it is much easier because of the use of the MPC; almost all of the necessary interface logic is buried in it. All that is needed is a little glue logic and buffering.

1.1 Overview

Before reading this document, the reader should have some exposure to the MULTIBUS II architecture, IEEE/ANSI 1296. Possibly this information has been gathered at a seminar or by looking at the bus specifications. Through either of these two means one might be aware of the parity bits that protect each of the various signal sets, arbitration logic, interconnect space, error indicators, etc. All of these features add up to a great deal of logic. If the control signals are fully understood, a fundamental read/write I/O board can be designed with an interface consisting of basic TTL and programmable logic devices. Since the signals are all synchronous to the bus clock and can easily be decoded it makes the design easy. However, designing a board with a microprocessor takes advantage of more advanced bus features and requires a little more work. The cost and real estate necessary to implement these features quickly make custom interface design prohibitive. The design of the MULTIBUS II Parallel System Bus, PSB, make it very reasonable to fabricate all bus functions in one VLSI device. The multiplexed signals, parity circuitry, arbitration logic, etc. are all interesting but unnecessary to deal with each time a board is being designed.

This application note will look at the various interfacing concerns of a CPU board's design and describe what the MPC provides and what the board designer needs to add.

1.2 Objectives

This application note is not intended to displace other documentation on the actual PSB bus operation or cover the internal operation of the MPC itself. The MPC interfacing required to connect to the PSB, the local CPU and the interconnect space capabilities of MULTIBUS II will be explained. The MPC is packaged in a 149-pin PGA (Pin Grid Array) package. In partitioning the pin functions of the MPC there are three major interfaces; the PSB interface, the local bus, and interconnect space (Figure 1). These are the activities to which most of the pins are allocated. The PSB interface is very straight-forward; there are few options for the designer. This makes that part of the board design simple. It also helps enforce compatibility with other boards. One has little latitude to make a mistake and develop an incompatibility problem.

The local bus interface of the MPC does offer the board designer a number of options. For instance, the local CPU might be 8-, 16-, or 32-bits. There are also DMA control lines and lines to assist in dual-port memory operation. Since the bus's design has its basis at Intel we would expect that the interface of the MPC to an Intel processor would be pretty well thought out, but what if a board design dictates a non-Intel processor? There are many boards being designed with other manufacturers' processors. For example, Huerikon Corp., Microbar Systems Inc., and Tadpole Technology all have developed CPU boards using the Motorola 68K microprocessor family.

The interconnect interface is intended to mate with the local bus of an 8051 series of microcontrollers. The 8051 coordinates the reset, configuration and diagnostics of the board. The design of interconnect space allows for some very powerful features to be designed into a board. With the capabilities of interconnect space it is possible to configure resources with software, enable or disable the board, provide imbedded diagnostics in the board, and at the very basic level provide machine readable information on the board's manufacturer, its part number and revision level. There is a companion application note (the MULTIBUS II Interconnect Design Guide, AP-423) that explores this subject in a lot of detail, so only the fundamentals of this MPC interconnect space interface will be described here.

After reading this application note, one should have a good basis for starting the detailed design of a unique board. Good Luck!

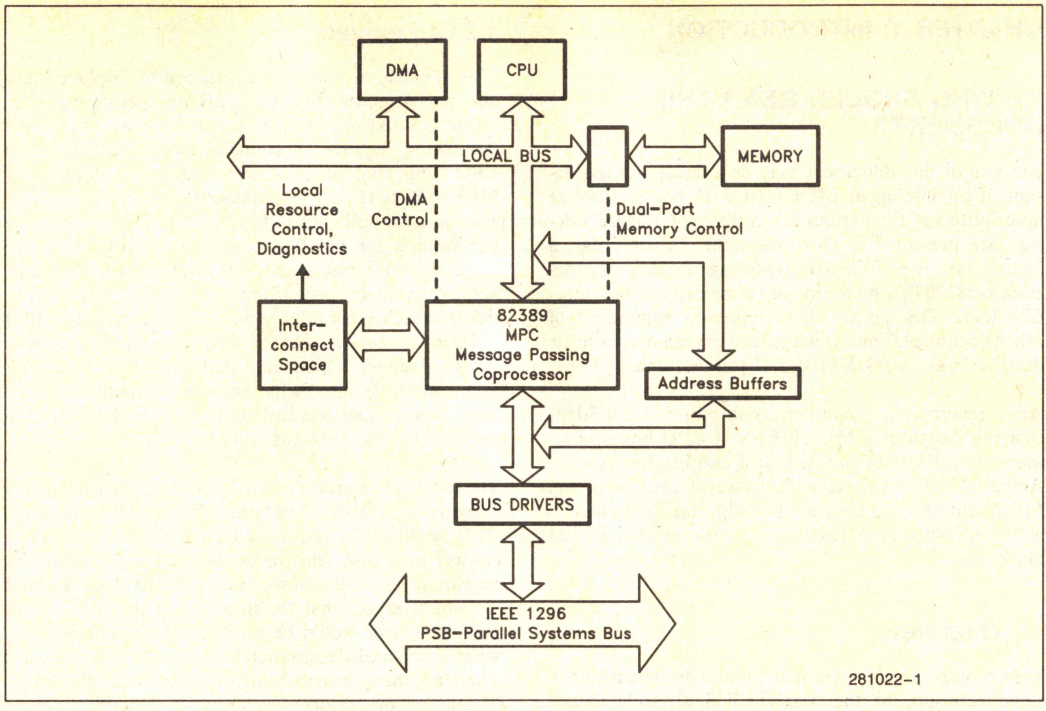


Figure 1. MPC Interfaces and Control Lines

CHAPTER 2: PSB INTERFACING

2.0 INTERFACING TO THE IEEE/ANSI 1296 PARALLEL SYSTEM BUS

All of the complex timing and control logic necessary to connect to the PSB are contained within the MPC. For the actual definition and operation of these control signals, refer to the IEEE/ANSI 1296 Specification. For the purposes of this application note, it is not necessary to understand all the signals, except in the case of reset and power fail situations, explained in section 2.2.2. There are a few mechanical orientation issues to be considered when laying out a board. These will be covered also.

2.1 IEEE/ANSI 1296 Fundamentals which must be Understood to Design a Board

The IEEE/ANSI 1296 description of MULTIBUS II defines very distinctly all the aspects of the PSB: the mechanical layout, the signal timings, signal drive capability, the bus protocol to other boards within the backplane, the backplane layout, etc. All boards are required to address the full 32-bit capability of the bus. Everything is well defined so that board designers will have few, if any, opportunities to develop incompatibilities between their boards and boards available from other vendors.

At a very fundamental level, the first thing that will impact design is the board size. The boards are based on the Eurocard format; the conventional size is 233.35mm (two connectors) x 220 mm deep. This is commonly referred to as a 6U board in the Eurocard style. This board format has two connectors, but all of the PSB requirements are satisified by one 96-pin DIN connector, referred to as P1 in Figure 2. P2 is uncommitted and allows flexibility for a unique design. This could include additional power requirements, specialized I/O, a subsystem interconnect bus, or whatever a unique design might require. For our purposes, P1 is the connector that we are interested in, so we will focus on the circuitry necessary to interface the MPC to the P1 connector.

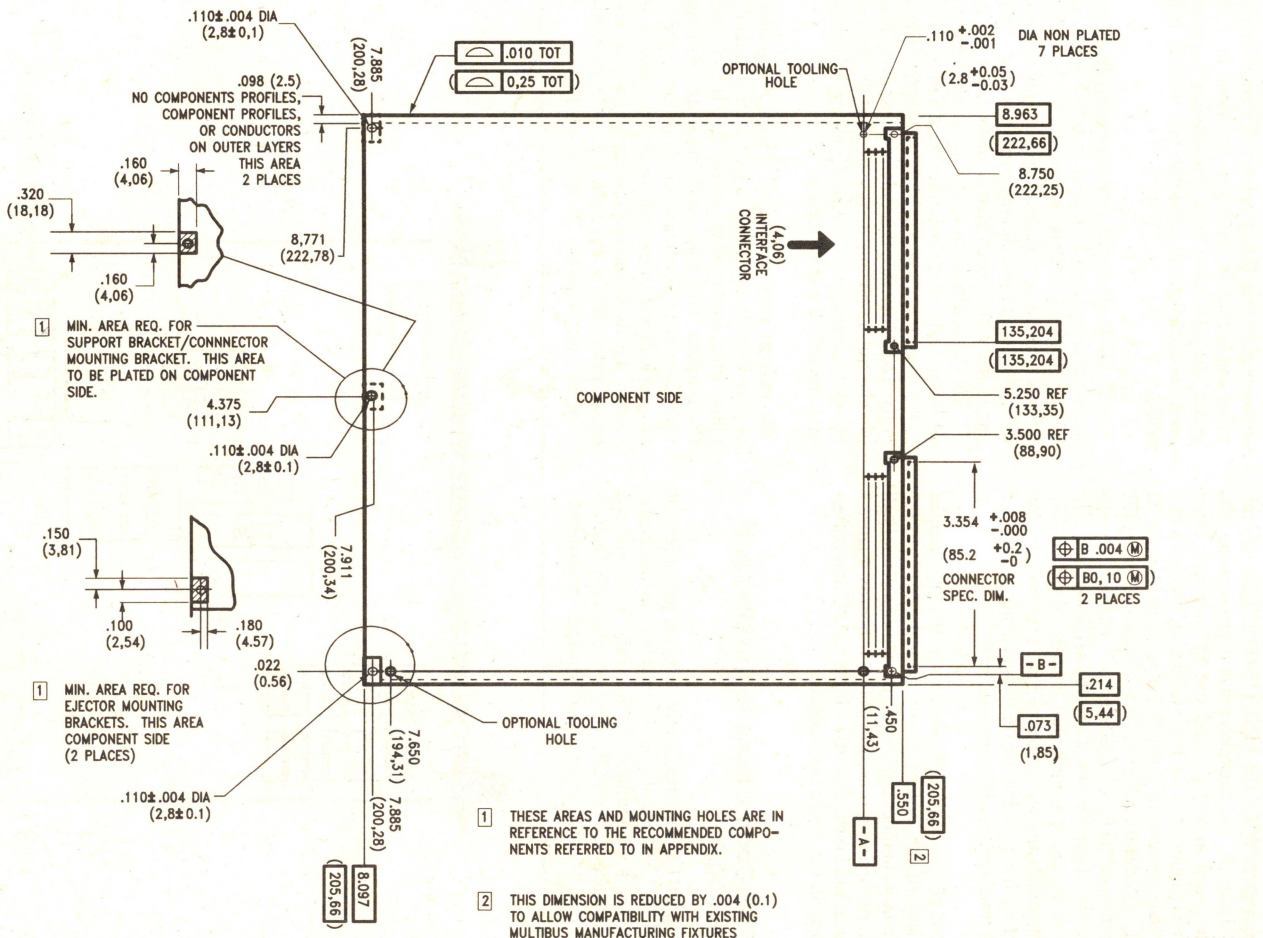


Figure 2. MULTIBUSII Board Dimensions

One subtlety to be accounted for here is the specification for the maximum trace length to bus driver at the P1 connector. When the MULTIBUS II PSB was initially designed, the backplane signal propagation was characterized so that the end system integrator could effectively ignore bus loading and timing problems in a backplane system that could range from two to 21 boards. The IEEE/ANSI 1296 specification defines the fabrication of the backplane to ensure reliable transmission of signals on the bus at the 10 MHz clock rates. In order to ensure that individual boards do not disturb the characteristics of the backplane, each board is required to have its bus driver components within 2.5 signal trace inches of the connector.

Because of the required dimensions to P1, and also the simplicity of the buffering logic from the MPC to the PSB, a large number of boards from various vendors all look very similar in this corner of their boards. Figure 3 shows an Intel iSBC 186/100 with the MPC and buffering logic in the lower right corner.

2.2 PSB Connections to the MPC and Local Logic

Since the MPC was specifically designed to implement the IEEE/ANSI specification in silicon, the interfacing requirements are minimal. Between the MPC and the P1 connection to the PSB, there are primarily bidirectional bus drivers, buffers, and straight connections. The address/data, parity, and system control lines all need bidirectional buffering. The MPC generates all of these signals and the necessary direction controls to point the buffers in the right directions in a transfer cycle. The arbitration lines, bus request, bus error, and reset-not-complete all drive the bus directly from the MPC.

The 32 BAD (Buffered Address/Data) lines each require buffering (even if an 8- or 16-bit CPU is used). All MULTIBUS II boards must decode the full 32-bit range of the memory address space. This eliminates all the possible address aliasing problems that can be encountered with boards that address less than the full 32 bits. It would seem that this would unnecessarily consume logic on boards that need far less than the 32-bit capability, however with the multiplexing of the entire 32-bit interface this only requires five TTL devices. The full 32-bit bus drive capability is also required to take advantage of message passing, which always uses the full 32 bits of the bus.

The current requirements of the bidirectional signals require from 48 mA to 64 mA of current sinking capability. Intel recommends using the 74F245 device for this purpose. This is an octal, bidirectional transceiver with three-state inputs and outputs. Just route the appropriate signals to the buffers with the appropriate direction signals. Seven of these devices are necessary to connect the MPC to the PSB. Figure 4 shows this buffering circuitry between P1 and the MPC.

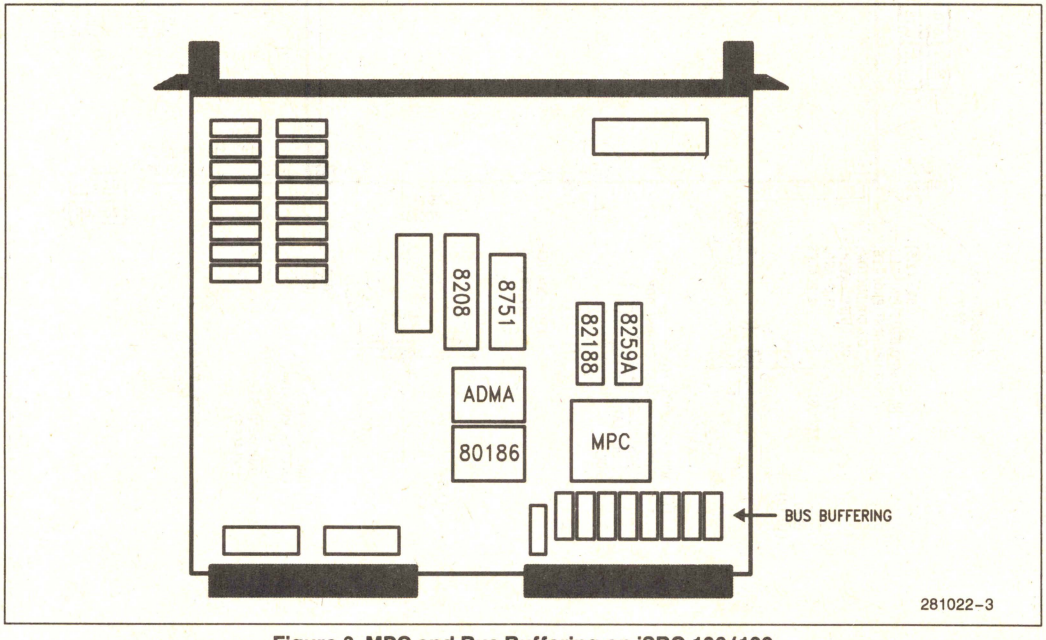


Figure 3. MPC and Bus Buffering on iSBC 186/100

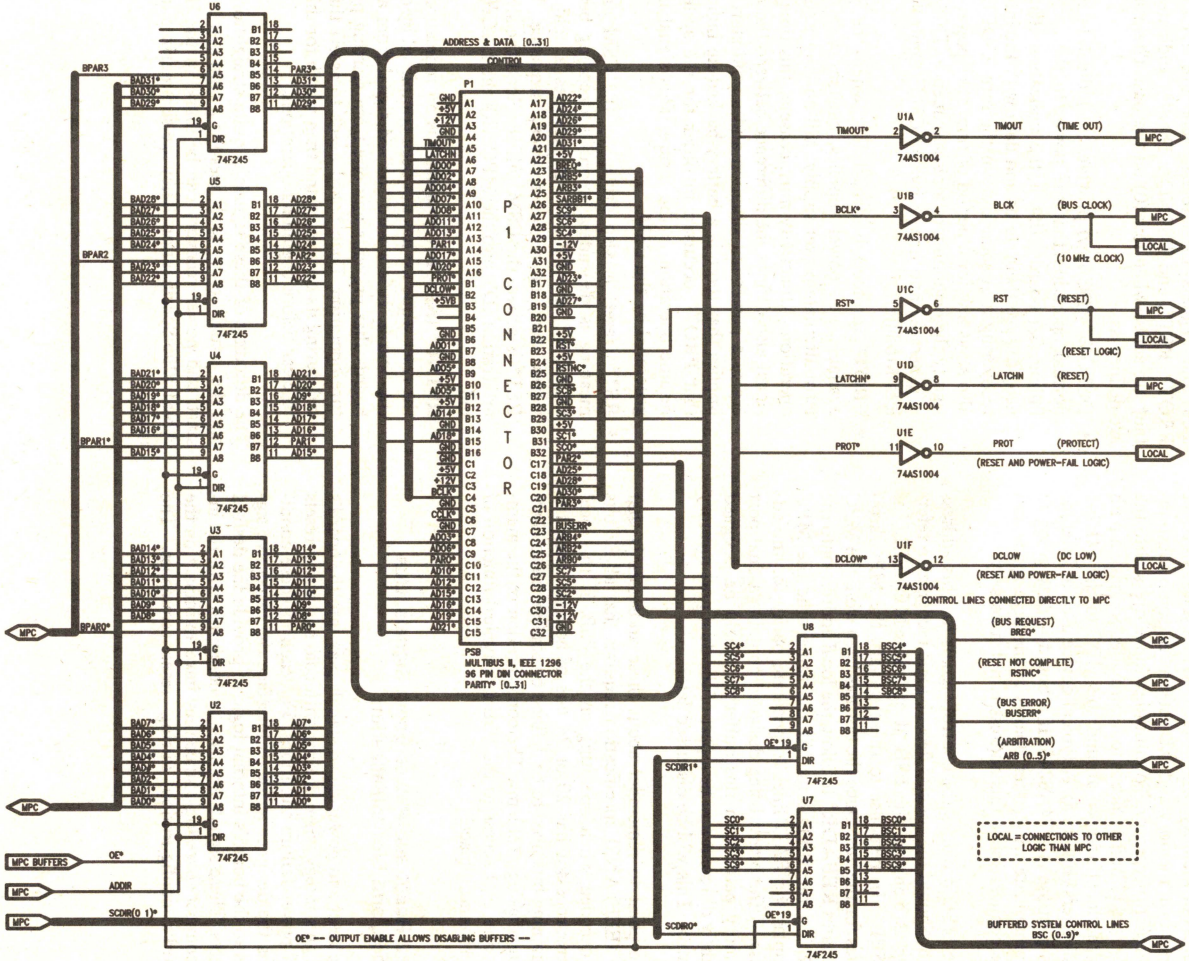


Figure 4. Circuitry between MPC and PSB

2.2.1 Providing PSB Address for Read, Write Cycles on PSB

Since the local bus interface is demultiplexed, the MPC does not provide a path to the multiplexed PSB for the address of memory or I/O read/write cycles. In order to perform memory and I/O references, some additional logic is necessary. To multiplex the address onto the bus, a set of buffers needs to be provided to make a path for the local address around the MPC to the 74F245 bus buffers (Figure 5). A typical device used at Intel for this purpose is a 74AS240. If a board is only going to communicate with other boards in message space and interconnect space this address is not necessary.

The MPC provides the control signals for this address path. Should the address decode logic recognize a reference as being on the PSB, it activates MEMSEL* or IOSEL*. In response to these signals, the MPC arbitrates for the bus. Once the bus has been won, the MPC activates REFADR* to allow the local address onto the PSB via the BAD lines. At the same time the MPC samples the BAD lines to generate parity for the address. If the board being designed contains a dual port memory path, then this buffering would be designed slightly different and be a part of the dual port memory control. This will be discussed later.

One point to remember about these addresses is that the IEEE/ANSI specification requires that all 32 bits be decoded when operating on the PSB. If the local microprocessor has 20 or 24 bits of address, then the remaining bits need to be defined or configured to eliminate aliasing problems. The configuration of the memory and I/O access to the PSB is a typical function of interconnect space.

2.2.2 Reset and Power-Down Signals, Bus Clock, and Slot Initialization

The reset and power-down signals play into the board's design strategy when dealing with cold and warm resets and power failure situations. For each of these conditions the designer needs to interface the respective signals to the board's internal logic to provide the desired effects. These signals are usually combined with the interconnect space logic since they have an impact on the board's configuration during reset sequences. The MULTIBUS II Interconnect Design Guide, AP-423, discusses this in more detail. For the purposes of this application note, the logical operation of these signals will be described.

The four signals of interest here are DCLOW*, PROT*, RST*, and RSTNC*. The "*" indicates that they are active when low (0) on the bus. These signals are defined as:

- **DCLOW* (DC low)** provides an indication during normal operation that the power to the system is falling. To boards, this is the indication that they need to save their status and data with the intent of recovering it at some time in the future. DCLOW* also stays low for a period of time after power-up to signify the beginning of a reset sequence. This signal is generated and driven by the CSM (Central Services Module).
- **PROT* (protect)** is driven low when the power available, i.e., power supplies, capacitors, or short term alternate power supply, will no longer support operation. All activity must cease at this time. This signal also defines the difference between the recovery of a power fail situation and a cold start. This implies that there is a battery-backed-up indication that PROT* occurred in order for the system to resume operation in the future. This signal is also generated by the CSM.
- **RST* (reset)** is also driven by the CSM and its meaning is dependent upon its decoding between DCLOW* and PROT*. It indicates a cold start, warm start, or power recovery sequence. While RST* is active, the CSM will also load the slot and arbitration ID's into the MPC with the LACHn signal.
- **RSTNC* (reset not complete)** is an indication that the reset sequence has not been completed by one of the boards. It is a wire "or"-ed signal from all the boards that prevents any one from starting premature operation. Each board needs to assert this signal during its own reset and then watch for this line to clear before beginning operations.

The decoding of these signals is quite straightforward, and the actual timing sequences for a cold start, warm start, and power recovery can be found in the IEEE/ANSI 1296 Specification. They are summarized in Figure 6.

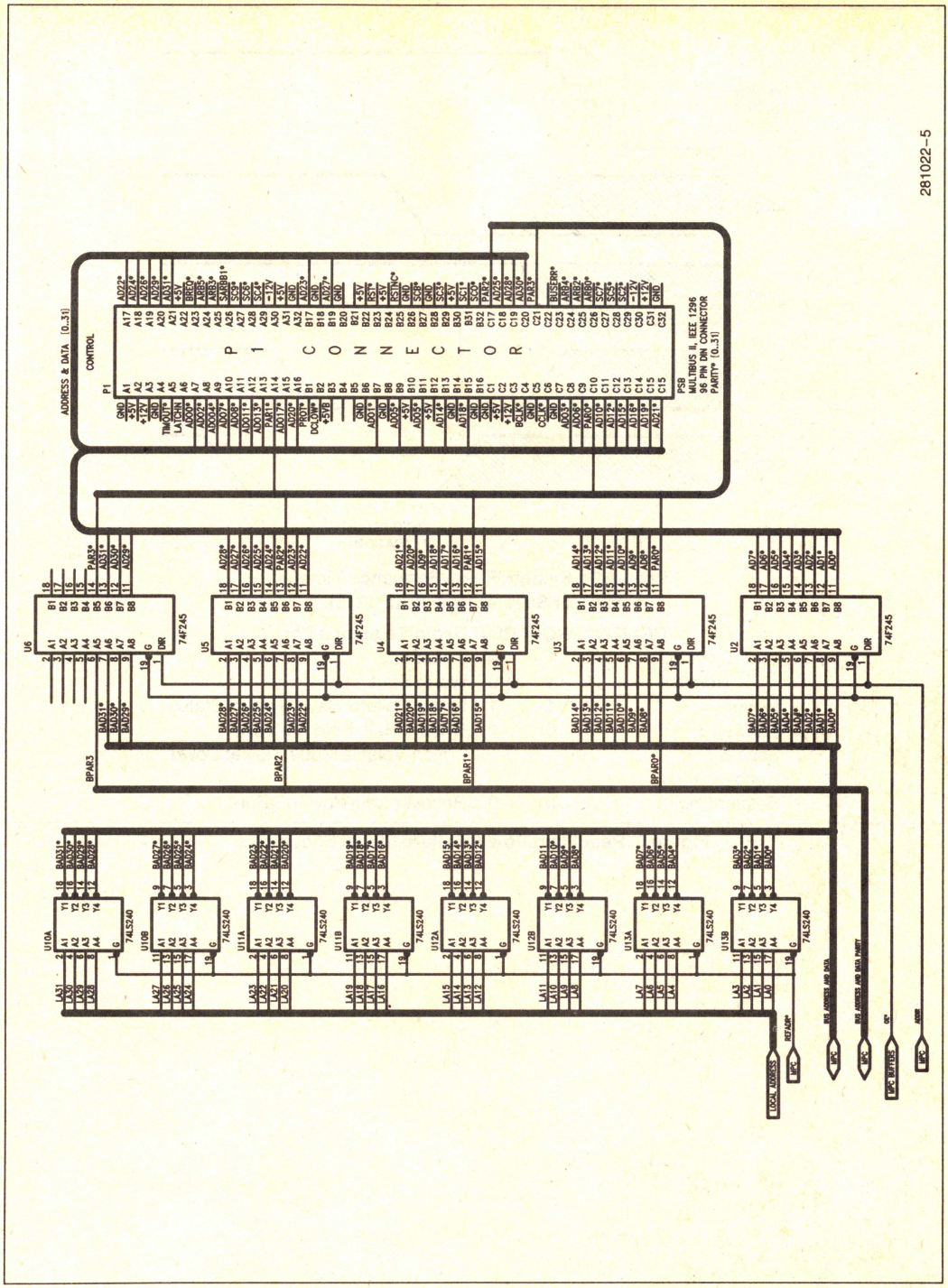
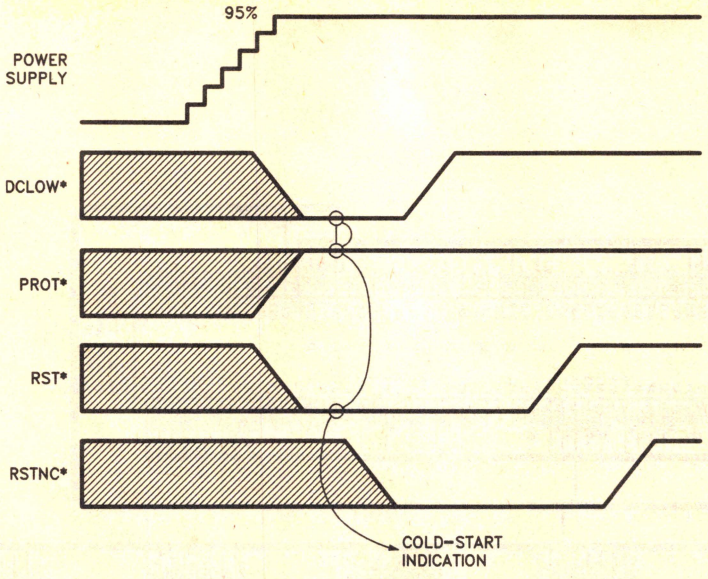


Figure 5. Providing PSB Address for Read/Write Cycles on the PSB



Power-On System Reset Sequence Timing
(One of Several from IEEE 1296)

	DCLOW*	PROT*	RST*	Begin Sequence For:
System Resets	0	1	0	Cold Start
	1	1	0	Warm Start
Power Failure Save & Recovery Sequencing	0	1	1	Power Failure, Save System Status
	0	0	1	System Halt, Hold Status in Non-Volatile Mode, Power Down
	1	0	0	Recover from Power Failure

Figure 6. Reset and Power Failure Signal Encodings

The RST*, DCLOW*, and PROT* signals typically are buffered by a 74AS1004 device. The RSTNC* signal is connected directly from the MPC to the PSB. RESET is carried from the RST* buffer directly to the MPC, then the three of these signals combined drive the reset control logic of the board.

There is another signal, LACHn*, which is driven by the CSM during the reset sequence. During the time that the RST* time is held low, the CSM is going through the backplane, loading the card slot and arbitration ID's into each MPC.

This LACHn line is also buffered by the same 74AS1004 device as the reset of the reset signals, and it is simply connected to the MPC. With this signal, the CSM is presenting the arbitration and card slot ID's to the MPC over the arbitration lines during the reset sequence. Within the backplane, each slot has its LACHn line connected to the unique AD (address data) line that is equal to the slot number in the backplane. The CMS pulls the appropriate AD line low and the LACHn line in that slot will go low, loading the ID's on the ARB lines. This is handled automatically by the MPC. You can refer to the IEEE/ANSI 1296 Specification for more details.

There are only two other signals which need buffering to the MPC. The first is TIMOUT*, which provides an indication to the MPC that a bus timeout has occurred. It is buffered by the 74AS1004 and it connects directly to the MPC. BCLK* is the second signal and it is also buffered by the same 74AS1004 and passed on to the MPC. This clock is used for all synchronous internal MPC timing.

At this point we have added one 74AS1004 to the seven 74F245 bidirectional buffers, to complete the connection of the MPC to the PSB. These buffering devices need to be connected as closely as possible to the PSB and MPC, so that the bus timing is not disturbed. Intel boards provide good examples of the layouts of these components.

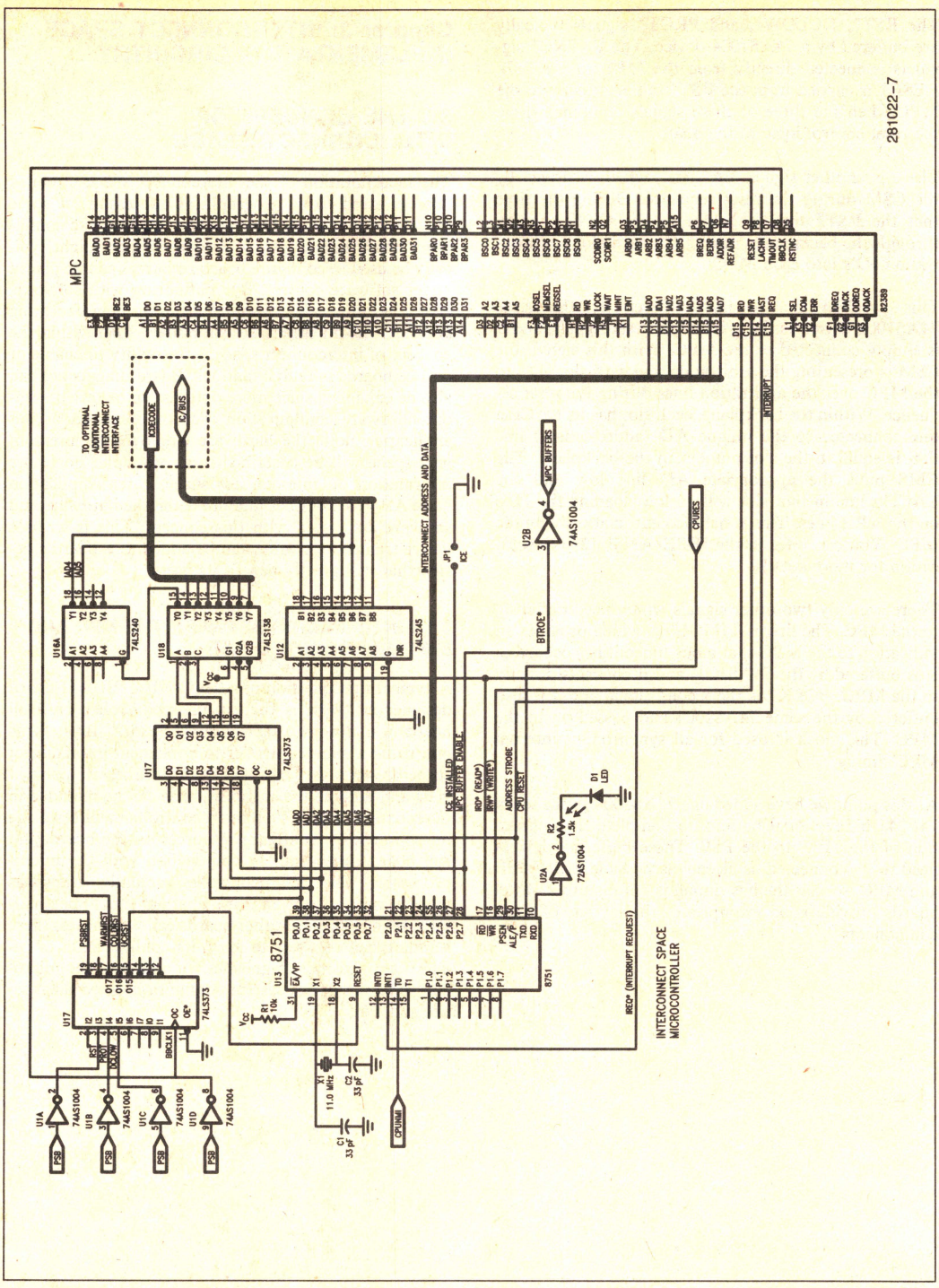
Chapter 3: INTERCONNECT SPACE IMPLEMENTATION CIRCUITRY

3.0 THE CONCEPT OF INTERCONNECT SPACE

The incorporation of the interconnect space circuitry (also referred to as geographical addressing in some buses) makes it easier for a systems integrator to configure a board into a system. Jumpers which might typically be used to configure board resources can be placed under software control. By configuring with software, errors in jumpers are eliminated during manufacturing and field modifications are simplified. Also within the concept of interconnect space is the ability to enable or disable boards in fault situations. Board diagnostics can also be enhanced since more options can be tested without hardware configuration. For the purposes of this application note, the hardware employed in interconnect space will be described in an example. For more information on interconnect space, Intel Application Note AP-423 describes in more detail the hardware and software associated with this concept. This is an area where tremendous value can be added to a board's design that will greatly benefit its users.

3.1.1 INTERFACING BETWEEN THE MPC AND 8751 MICROCONTROLLER

A typical first question of a new MULTIBUS II board designer is "Why is there a microcontroller on each board in addition to the microprocessor?" As it turns out, using a microcontroller to implement interconnect space becomes very efficient from a real estate point of view and adds to the capability that can be included in interconnect space. This microcontroller will reset the microprocessor, configure resources on the board, and run board diagnostics in conjunction with the microprocessor. The microcontroller actually translates many complex functions into what appears to be a uniform set of registers. Other boards can then access these standardized registers to learn its configuration or request it to execute diagnostics. In addition, the microcontroller simplifies system configuration and diagnostics software.



281022-7

Figure 7. 8751—MPC Interconnect Space Circuitry Example

The MPC is designed to connect directly to the data bus of an 8751. Figure 7 shows typical 8751-MPC interconnect circuitry. The address/data bus, labeled IAD <0...7>, is used to transfer data and addresses between the 8751 and the MPC. The data and address transfer operations are controlled by the control lines labeled read, write, strobe, and interrupt. A typical interconnect operation is triggered by the MPC recognizing an interconnect space address from the local CPU or the PSB. The MPC then sends an interrupt request to the microcontroller telling it that an interconnect access needs to take place. After the interrupt request, the microcontroller reads the address and data off the address/data bus, then executes the appropriate interconnect service routine.

Figure 8 shows a diagram for interconnect bus timing. Both a read and a write diagram are presented. In both the read and write operations the microcontroller sends the IAST (strobe) signal to the MPC telling it that there is a valid address on the bus. The MPC will then latch the address into its buffer. At this point, when a read operation needs to occur, the microcontroller will send an IRD* (read) signal to the MPC, signalling that it wants to read data from the address specified. The MPC will then supply data to the microcontroller from the specified address. Conversely, if a write operation needs to occur, the IWR* (write) signal is sent to the MPC indicating that the microcontroller wants to send information to the specified address via the MPC. The MPC will then proceed to receive data from the microcontroller and send it to the specified address.

3.1.2 CONTROLLING BOARD RESOURCE CONFIGURATIONS THROUGH INTERCONNECT SPACE

Board designers will configure local board resources through interconnect space. Figure 7 indicates an additional branch off the address/data bus. This branch is the address/data bus to other configuration devices, or status indicators on the board. In addition, LIAD0.2 of the bus branches to a decoder which will coordinate selection of the particular device. The one path shown to U16A allows the 8751 to read whether a warm or cold reset is occurring. More details of these operations can be found in the MPC User's Manual, 286/100A User's Guide, and Application Note AP-423.

3.1.3 RESET OPERATIONS

The reset PAL is used to coordinate the reset operations. Figure 7 displays the control signals for the reset PAL and the associated output of the PAL. Reset signals are received over the PSB and decoded through the reset PAL. The reset PAL is able to generate a cold-reset or a warm-reset. The reset PAL also resets the 8751, which starts it through its reset sequence. One of the first things it determines is what type of reset is occurring. When address/data lines <0–2> are all low, it causes a 3 x 8 decoder to enable the U16A buffer, putting the cold or warm reset signal on the bus of the 8751. Once on the address/data bus, the microcontroller and MPC read this signal and execute the appropriate reset sequence.

The condition of a cold start reset causes all functions of the system to act as an original power-on condition. Furthermore, the MPC and interconnect registers are set to their default values, and any initialization checks will occur. A warm start is usually caused by the human operator pressing the reset button while there is power to the system. The sequence of events are the same for a warm and cold start, however, a warm start will not modify interconnect registers or memory, and initialization checks are not executed.

5

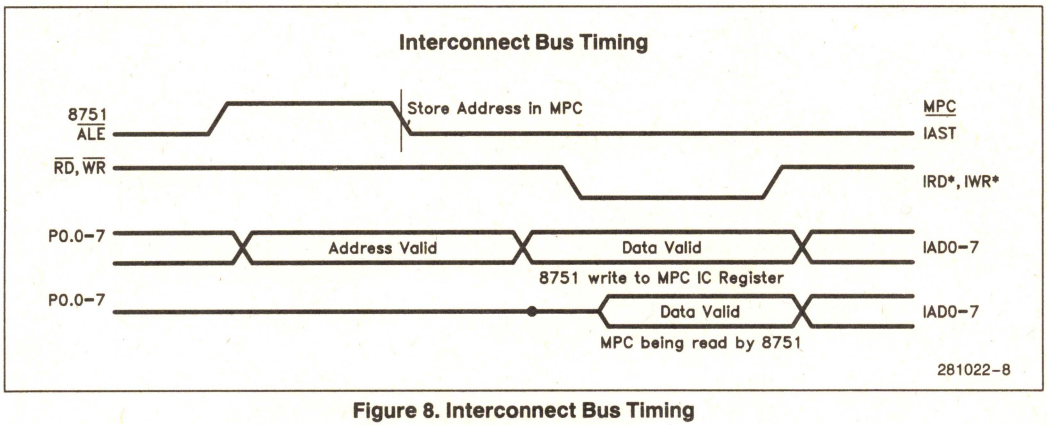


Figure 8. Interconnect Bus Timing

CHAPTER 4: INTEL MICROPROCESSOR INTERFACING EXAMPLES

Interfacing a microprocessor to the MPC is very similar to the interface that occurs typically between a microprocessor and a peripheral device. The notable differences, in addition to the fact that it has a lot of registers, are that the MPC:

- Supports a 32-bit data path,
- Provides support for generic microprocessor signals, lock, mem, and I/O sel, etc. to pass through to the PSB,
- Has byte-enable lines allowing 8-, 16-, or 32-bit microprocessors of different architectures to interface to the MPC and the PSB.

Examples of some interface possibilities are described for various Intel microprocessors in the subsequent sections.

4.1 80186 Interface Example

The signals required to interface the 80186 to the MPC are all shown in Figure 9.

Starting at the top of the MPC, the byte-enable lines (BE) are set for a 16-bit microprocessor; BE2 and BE3 are tied high to allow the microprocessor to address the MPC registers or the PSB via the two lower bytes, D0–D15, of the data lines. A0, *BHE, allow the 80186 to transfer 8 or 16 bits of data appropriately across D0–D15. Both the 80186 Hardware Reference Guide and the MPC User's Manual describe these signals in more detail.

The data lines AD <0...15> map from the 80186 to the MPC. The data on these lines is appropriately timed with the read and write lines of the 80186, however the 80186 multiplexes address and data and the appropriate address for the MPC register needs to be picked off these lines with the ALE signal. The 74L573 is recommended for this function.

The function of the Decode Logic element is to capture the address coming out of the 80186 and, using the 80186 status signals, decide if the address is a PSB I/O (IOSEL), PSB Memory (MEMSEL), or MPC Register (REGSEL). The read and write signals would drive the data to its appropriate destination.

The signal (LOCK*) locks the bus from other microprocessors during semaphore operations and (WAIT*) holds off the 80186, until the MPC and the addressed resource on the bus are ready to be read or written.

Two interrupts can be generated by the MPC: MINT will go active to the CPU if there is an outstanding message operation. EINT is asserted if there are errors which occurred in memory, I/O, or Interconnect space. Both of these would require software routines to appropriately handle the situation.

The 80186 possesses the ability to perform DMA operations. If cost must be kept to a minimum and/or real estate is very limited, the use of a separate DMA controller, therefore, is not necessary if the 80186 is implemented. In order for the 80186 to accomplish DMA, DRQ0 and DRQ1 of the 80186 must be tied directly to the DMA request lines of the MPC. Each request line of the 80186 controls a DMA channel. In Figure 9, channel 0 of the 80186 is used for DMA input to the MPC and channel 1 is used for the output to the 80186.

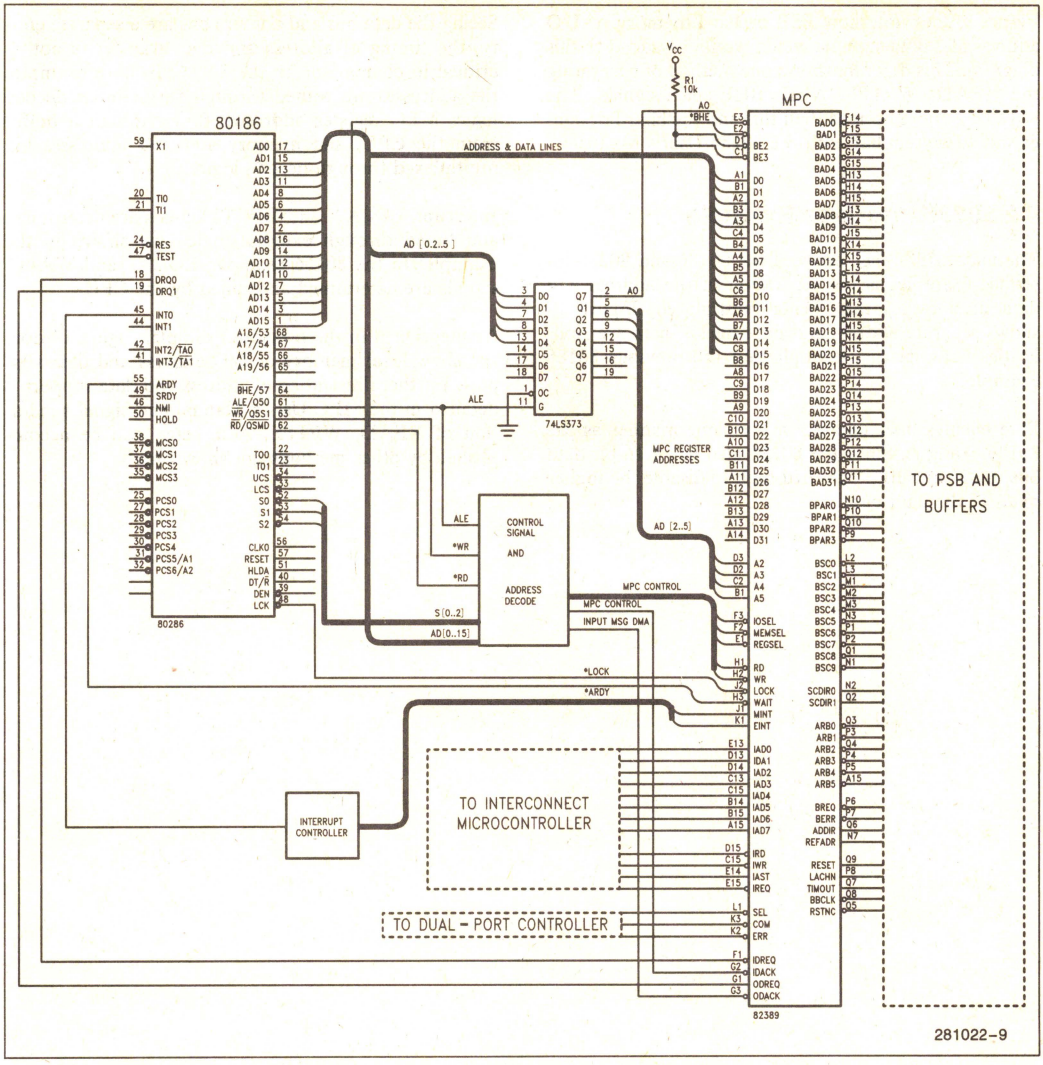


Figure 9. 80186—MPC Interface Example

Figure 9 does not show PSB or local memory or I/O address and data support which would be added to this diagram. This diagram shows one method of generating the READ, WRITE, ALE, BHE, etc. signals. The 80186 has more options than this and the board designer should be familiar with the 80186 User's Manual.

4.2 80286 Interface Example

The major difference between the 80186 and 80286 interface examples is the separation of the address and data lines rather than being multiplexed. Figure 10 shows separate address and data lines. There are similarities between this example and the previous 80186 example.

Byte enables lines operate in the same manner as the 80186 example. Since the 80286 also has a 16-bit data bus, BE2* and BE3* are tied high to disable the higher bytes of the data bus.

Seeing the data bus and address bus are a separate entity, the timing of address and data transfer is not as critical in comparison to the 80186. In both examples the addresses are routed through very similar decode logic; MPC register addresses flow through a buffer from the CPU, and memory and I/O addresses are multiplexed through decode logic.

Interrupt, LOCK*, and WAIT* signals are determined and timed through external logic now shown in the example. In the 80186 example, LOCK* and WAIT* signals are determined and timed by the CPU.

As mentioned in the previous example, Figure 10 does not show local memory or I/O address and data support. Further illustration of address and data support is detailed in the MPC User's Manual and signal generation of READ, WRITE, ALE, etc., can be accomplished by other methods not shown here.

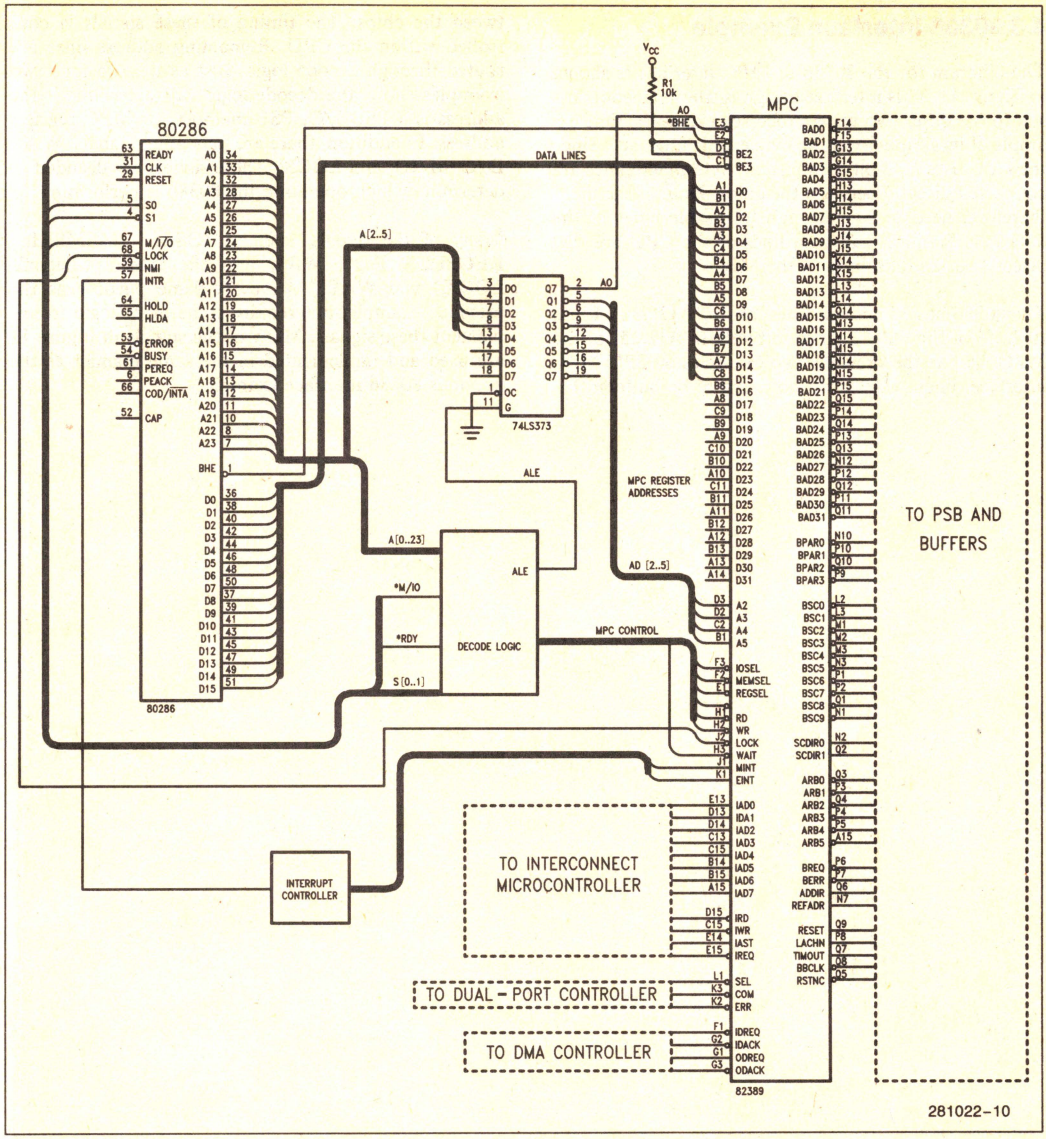


Figure 10. 80286— MPC Interface Example

5

4.3 80386 Interface Example

The diagram for the 80386 to MPC interface is shown in Figure 11. This interface design is simple in comparison to the other two microprocessor examples. One example of its simplicity is the byte enable interface. Since the CPU uses a 32-bit data bus, all byte enable lines are connected directly between the chips. Note of the lines, therefore, need to be tied high to disable bytes of the data bus. Similarly, all data lines of the CPU are tied directly to the data lines of the MPC.

The 80386 has 30 address lines labeled A[2...31]. Lines A[2...5] of the CPU connect directly to A[2...5] of the MPC. Unlike the previous two examples of CPU MPC interface, lines A[2...5] do not need to be buffered between the chips. The timing of these signals is controlled within the CPU. Remaining address lines are routed through decode logic. Just as the previous two examples show, the decode logic will determine if the address is a PSB I/O, PSB memory, or MPC register address. In addition, there are four other signals: W/R, D/C, M/IO, and LOCK. These signals are decoded to determine which operation the 80386 is performing.

Many of the signals, such as IOSEL*, MEMSEL*, REGSEL*, RD*, WR*, and the interrupt signals LOCK* and WAIT* serve the same purpose as the previous example, but require some extra logic to implement these signals. Many of the signals in Figure 11 are used and implemented in the same manner as the previous 80286 interface example.

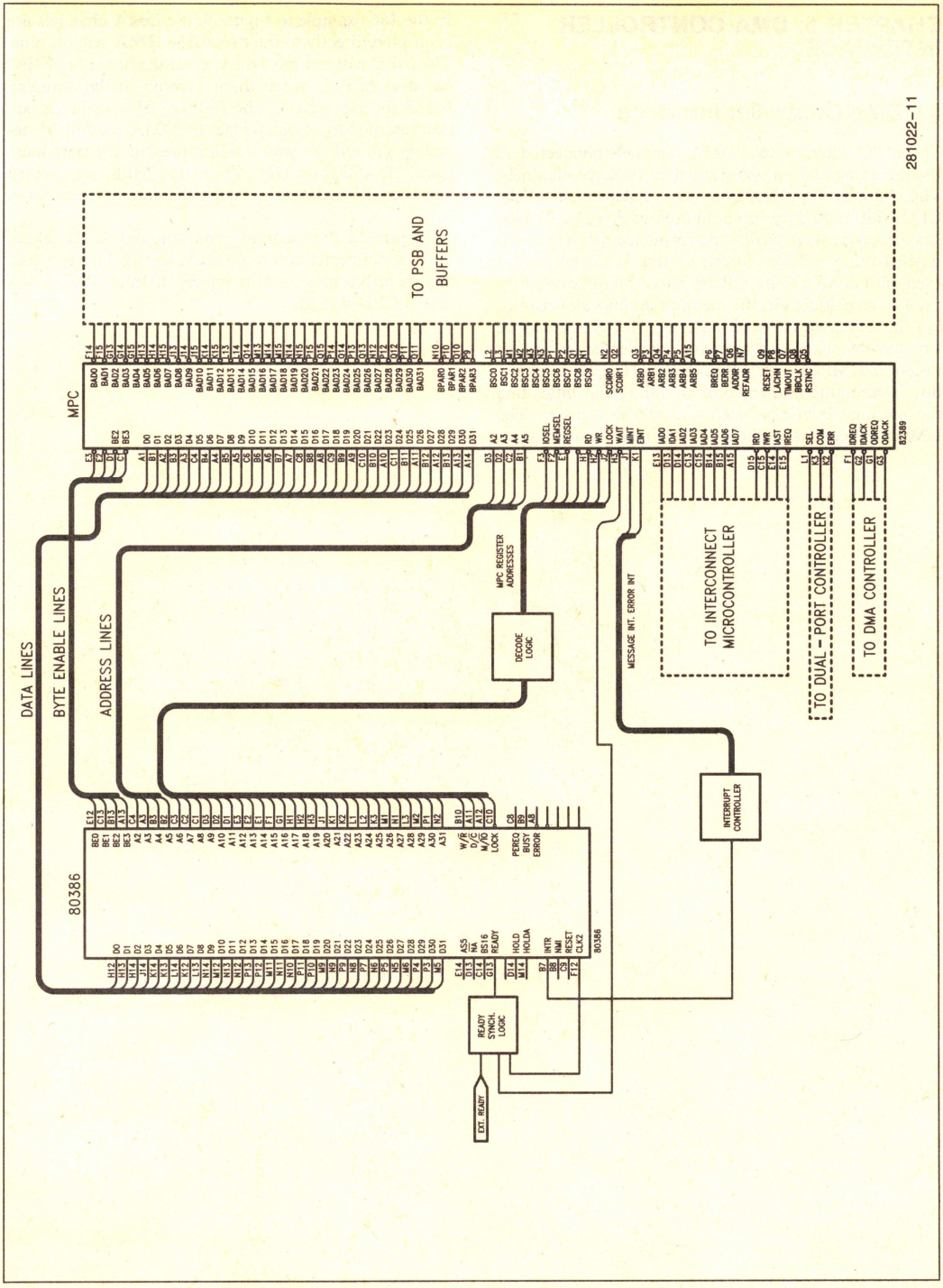


Figure 11. 80386—MPC Interface Example

CHAPTER 5: DMA CONTROLLER INTERFACE

5.0 DMA Controller Interface

The MPC contains two DMA channels connected to the incoming and outgoing solicited message channels. The width of the DMA channel is independent of the CPU width and may be configured as 8, 16 or 32 bits. Maximum message passing performance will be accomplished using a 32-bit interface; this is recommended even with 8- and 16-bit CPUs. The DMA address may need to be shifted and the memory addressed in banks, but the increased message passing performance may be worthwhile. Once a large data transfer is set up between two MPCs over MULTIBUS II, a DMA function is recommended to continue filling and unloading the message FIFOs at each end of the transfer. The DMA connections to the MPC are really quite simple.

In the 186 example in Figure 9, the DMA channels are connected directly to the two 80186 DMA request pins. The MPC will request DMA when the incoming FIFO has data in it or when there is room in the outgoing FIFO for more data. The 80186 DMA channels are programmed to acknowledge the DMA request by decoding an address which will represent the port locations. IDACK and ODACK of the MPC are enabled and the CPU will read or write the data to these ports.

In Figure 12 the incoming and outgoing DMA channels are connected to the ADMA, 82258. This is a very simple interconnect and is typical of Intel's 80286 and 80386 CPU boards.

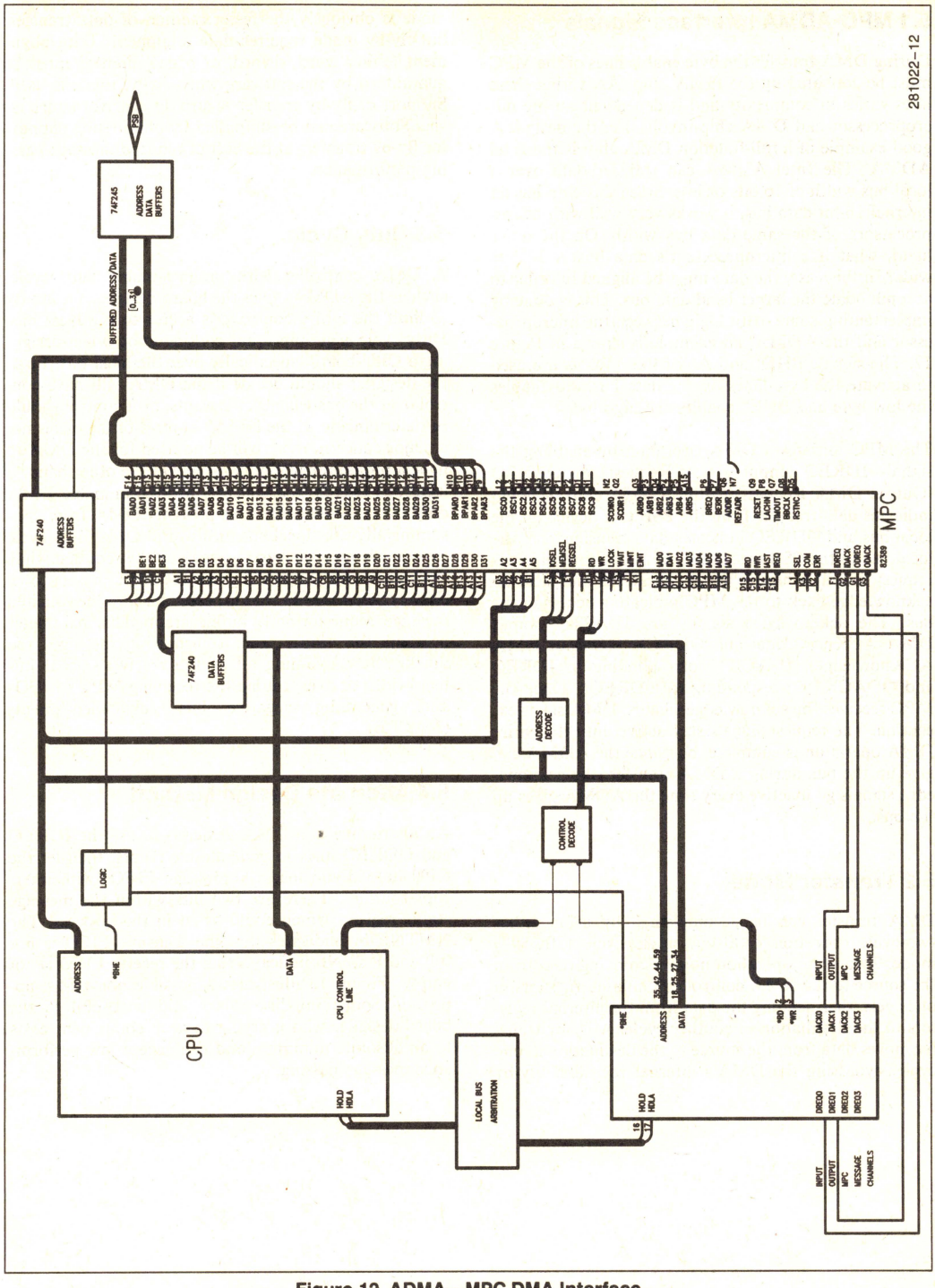


Figure 12. ADMA—MPC DMA Interface

5.1 MPC-ADMA Interface Signals

During DMA transfer the byte enable lines of the MPC must be activated by the DMA chip. Activating these lines varies in complexity and is dependent on the microprocessor and DMA chip involved in the design. A good example of a full-function DMA chip is the Intel ADMA. The Intel ADMA can transfer data over a local bus width of 16 bits or less. Since this chip has an internal 16-bit data bus, it works very well with microprocessors of the same data bus width. On the other hand, what if a microprocessor's data bus is 32 bits wide? In this case, the data must be aligned in order to accommodate the larger local data bus. This is done by implementing some extra logic between the microprocessor and the ADMA; an example is shown in Figure 12. The signals BHE* and A0 of the ADMA are used to activate the byte lanes on the data bus. A0 enables the low byte and BHE* enables the high byte.

The MPC initiates a DMA operation by enabling the signals IDREQ (Input DMA Request) or ODREQ (Output DMA Request) to the ADMA chip. IDREQ indicates data transfer from the PSB to a device on the local bus and ODREQ indicates data transfer from the local bus to the PSB bus. Once the ADMA receives its request, it will arbitrate for the local bus and send an acknowledge back to the MPC indicating access to the bus. The acknowledge signals are IDACK* (Input DMA ACKnowledge) and ODACK* (Output DMA ACKnowledge), IDACK* corresponding to IDREQ and ODACK* corresponding to ODREQ*. Once the MPC receives the acknowledge signals, DMA starts execution. The request signals stay active until the entire DMA operation is complete, but since the ADMA can give up the bus during a DMA transfer, the acknowledge signals go inactive every time the ADMA gives up the bus.

5.2 Transfer Mode

DMA transfer can be accomplished by using either two-cycle operation or using single cycle ("fly-by") mode. Two-cycle operation involves moving data from the source to the DMA controller's internal registers in one cycle, then sending the data from the internal registers to the destination in another cycle. A fly-by transfer moves data from the source to the destination in one cycle, bypassing the DMA's internal registers. Fly-by mode is obviously the faster method of data transfer, but fly-by mode requires data alignment. Data alignment (either word, dword, or paged aligned) must be guaranteed by the software when fly-by mode is used. Support of fly-by transfer is strictly a performance issue. Software can be simplified by eliminating support for fly-by transfers at the cost of reduced message passing performance.

5.3 Duty Cycle

A DMA controller with programmable duty cycle, such as the ADMA, gives the board designer the ability to limit the DMA controller's access to the local bus. Duty cycle is an important consideration when designing a CPU board, specifically over the local bus. First, the designer should decide if the DMA will have control over the bus whenever it wants, or if control should be deterministic. If the DMA's control of the bus is not a major concern, there will be no need for the DMA to have a programmable duty cycle. On the other hand, if local bus latency is a concern, a programmable DMA duty cycle is desirable. This prevents DMA from consuming all of the local bus bandwidth. DMA duty cycle (to throttle message passing) may not be required when designing with the MPC since the MPC has its own duty cycle parameter that is passed back to the solicited message requestor in a buffer grant. This parameter prevents the sender from transmitting messages too quickly by consuming all of the receiver's local bus bandwidth or overrunning the receiving MPC's FIFOs and generating negative bus acknowledgements (NACKS).

5.4 Alternate Design Method

An alternative to a DMA device is to use the IDREQ and ODREQ lines to generate interrupts, then let the CPU unload and load the message FIFOs with interrupt routines. There are two bits within the message status register which could assist in this process. The XMTNF bit indicates that the transmit FIFO is not full and RCVNE indicates that the receive FIFO is not empty. Figure 13 illustrates a possible non-DMA approach. Something like this would be useful to the board designer who is trying to keep component costs to an absolute minimum and can accept low performance message passing.

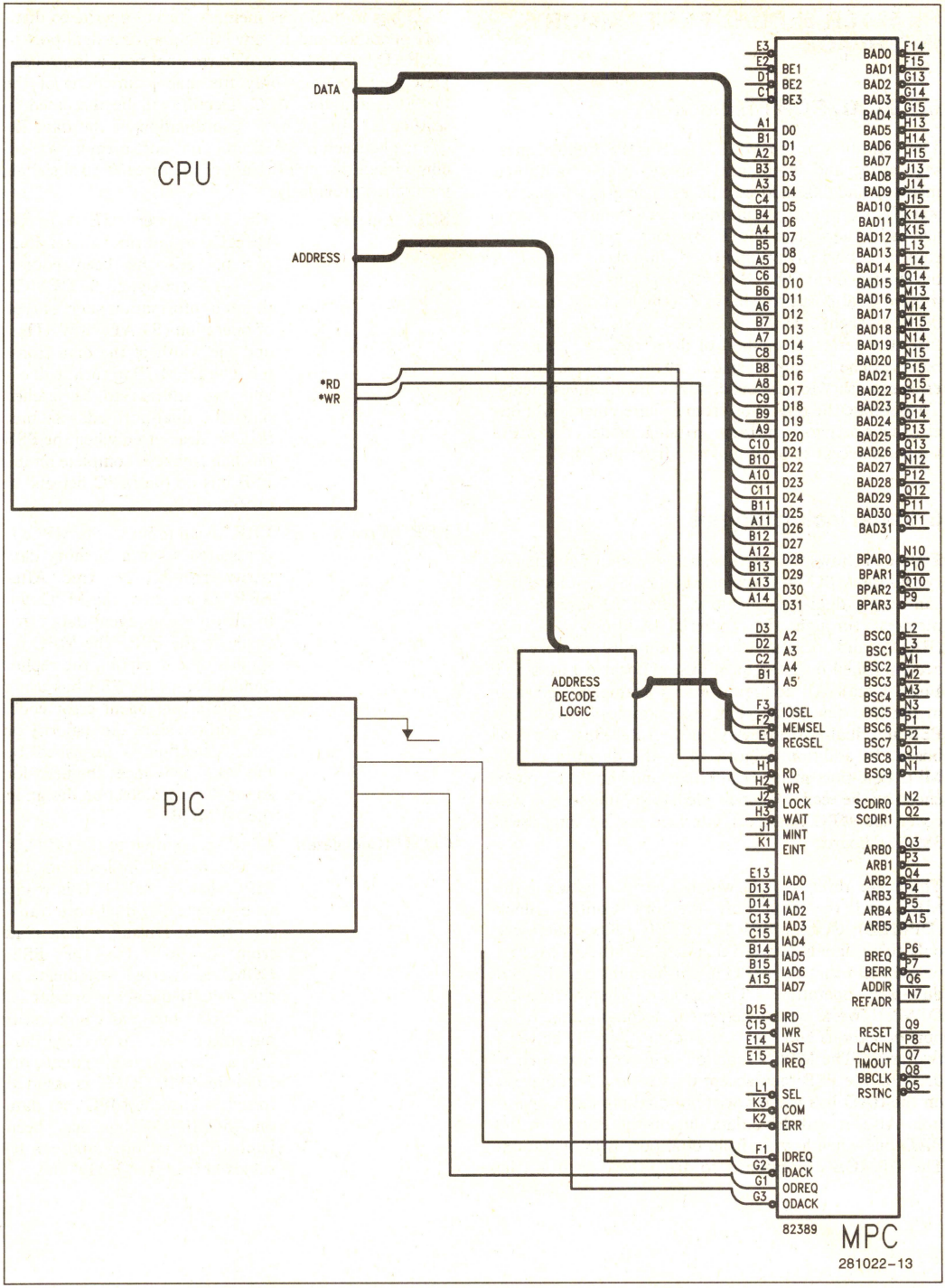


Figure 13. Servicing Message FIFOs without a DMA Controller

5

CHAPTER 6: DUAL-PORT MEMORY INTERFACE

6.0 DUAL-PORT MEMORY

The definition of the MULTIBUS II PSB added message space and interconnect space to the traditional memory and I/O address spaces. The goal of message space is to support standardized data transfer between agents on the system bus (PSB). Each agent's processor runs in its own environment and controls its own memory management. The added message space and already present dual-port provide two methods of data transfer between agents. Under the MULTIBUS II architecture, the perferred method of data transfer is message passing. The traditional shared memory model, however, is necessary to support certain software investments. To facilitate the migration from a shared memory model (dual-port) to a message passing model, the MPC supports access to local memory from the PSB.

6.1 Interfacing Signals

Figure 14 shows the interface between dual-port memory and the MPC. In this diagram the PSB and local bus can access dual-port memory. The PSB accesses the memory through the Buffered Address/Data bus (BAD*) and local accesses to dual-port memory are accomplished over the local buses (Local Address and Local Data). All communications examples discussed in the next few paragraphs are examples between the PSB and dual-port memory on the Intel iSBC 386/100 board. In addition, Figure 14 is the diagram of the MPC dual-port memory interface and should be referenced by the reader when needed. Also, this section will cover the MPC dual-port interface on the Intel iSBC 386/116 board.

There are three signals which interface between the MPC and the Dual-Port Memory Control Block (DPMCB): COM*, ERR* and SEL*. Since there are only three signals between the MPC and the DPMCB it would seem that the MPC might have little to do with dual-port operations. The fact is, the MPC and DPMCB work well together in accomplishing dual-port operations and the work is divided evenly between two areas. The MPC's function is to recognize the address on the PSB then accept data from it and drive it on the BAD bus when a write to dual-port is in operation. Also, it will drive data out on the PSB from the BAD bus when a write from dual-port is in operation. The DPMCB's function is to drive the data from the BAD bus to dual-port memory during a write to dual-port operation and to drive the data from dual-port to the BAD bus when a write form dual-port is in process. Described above is only the macro functions of the DPMCB and the MPC. Details will be discussed in section 6.2. In addition, coordination of the data exchange between the PSB and dual-port memory are accomplished through the use of the three control signals mentioned previously.

SEL* (Select): The MPC asserts SEL* to the DPMCB to inform it that a dual-port operation has been requested. SEL* will signal the DPMCB to latch information such as type of operation (READ or WRITE) and the width of the data transfer. The DPMCB in turn, will enable the address to be latched onto the dual-port address bus. SEL* is deasserted when the PSB bus handshake is complete on the PSB bus or the MPC detects an exception.

ERR* (Error): ERR* is an input to the MPC. It is asserted when a memory data parity error has occurred. After ERR* is asserted, the MPC will in turn place an agent data error signal to the PSB. The MPC responds by completing the replier handshake on the PSB bus using a "data error" agent error code. In addition, since the majority or error detection is furnished by the MPC, it reduces the need for so much error detection design in the DPMCB.

COM* (Complete): COM* is an input to the MPC. It is asserted to indicate to the MPC that the DPMCB is ready to complete the dual-port transfer process. During a data read from dual-port to the PSB, COM* is asserted to indicate to the MPC that data is present on the BAD* bus and can be sent out to the PSB. Conversely, during a data write to dual-port from the PSB, COM* is asserted to indicate to the MPC that data on the BAD* bus has been latched into memory and can be removed from the BAD* bus.

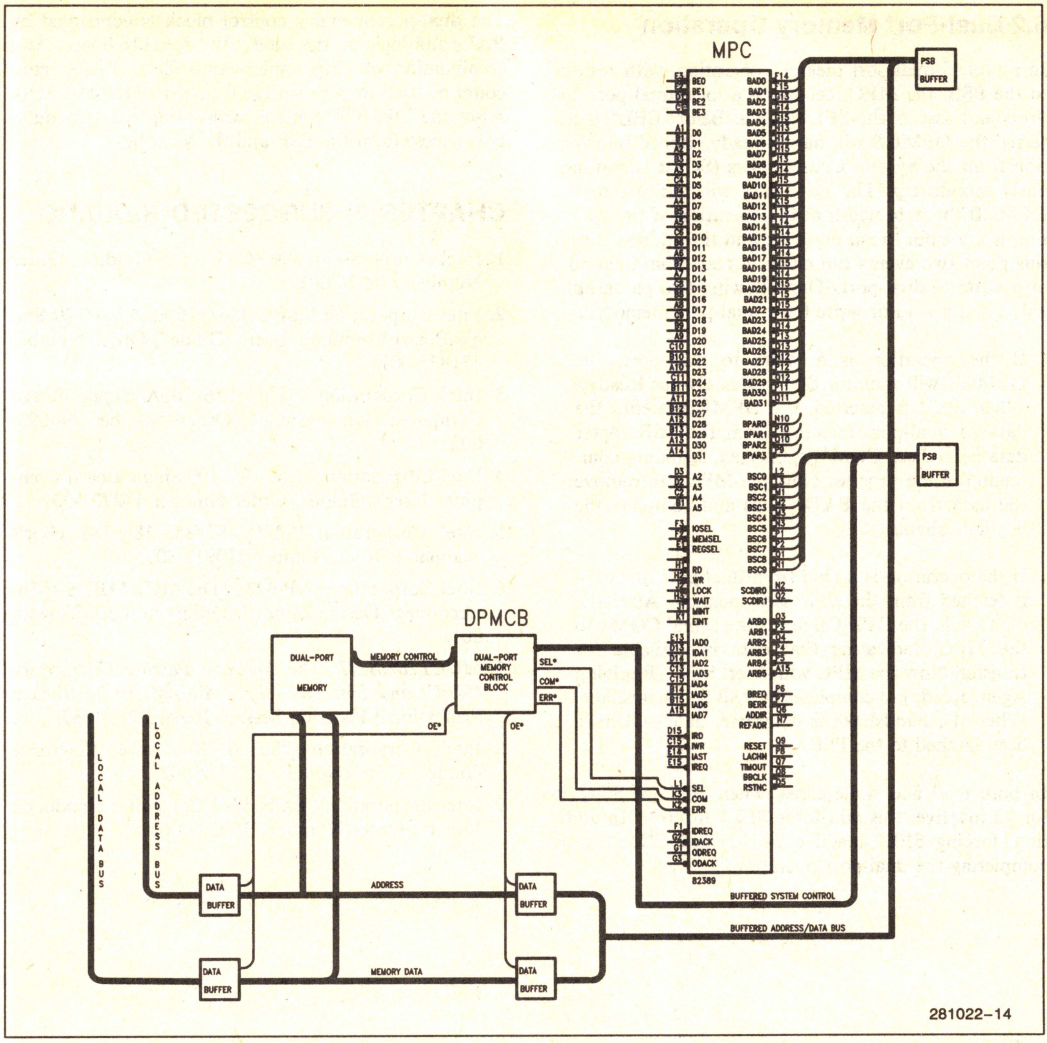


Figure 14. MPC Dual-Port Interface

5

6.2 Dual-Port Memory Operation

In a typical dual-port memory operation, with respect to the PSB, the MPC recognizes a local dual-port address and asserts the SEL* signal. Before SEL* is asserted the DPMCB will have already latched information from the System Control lines (SC) it is continuously monitoring. The BSC lines will indicate to the DPMCB the data width of the operation, if the operation is a memory read or write, and the address. From this point two events can occur: a read from dual-port or a write to dual-port. The following two paragraphs will detail a read or write from dual-port memory.

If the operation is a write to dual-port, the DPMCB will monitor SC3* (Bus Owner Ready). When SC3* is asserted, the DPMCB latches the data to dual-port memory from the PSB. After data is latched, COM* is asserted, signaling completion of the process. Now the MPC can remove the data from the BAD* lines and complete the bus handshake.

If the operation is a read from dual-port, the data is fetched from the addressed location. After the data fetch, the DPMCB sends the signal COM* to the MPC, indicating that data is available for transfer. Now the MPC will assert SC4* (Replying Agent Ready) to complete the PSB bus handshake. When the handshake is complete, data will have been latched to the PSB.

In both read and write cases, when SC3* or SC4* is forced inactive, this will force SEL* inactive. In addition, forcing SEL* inactive will disable COM*, thus completing the dual-port operation.

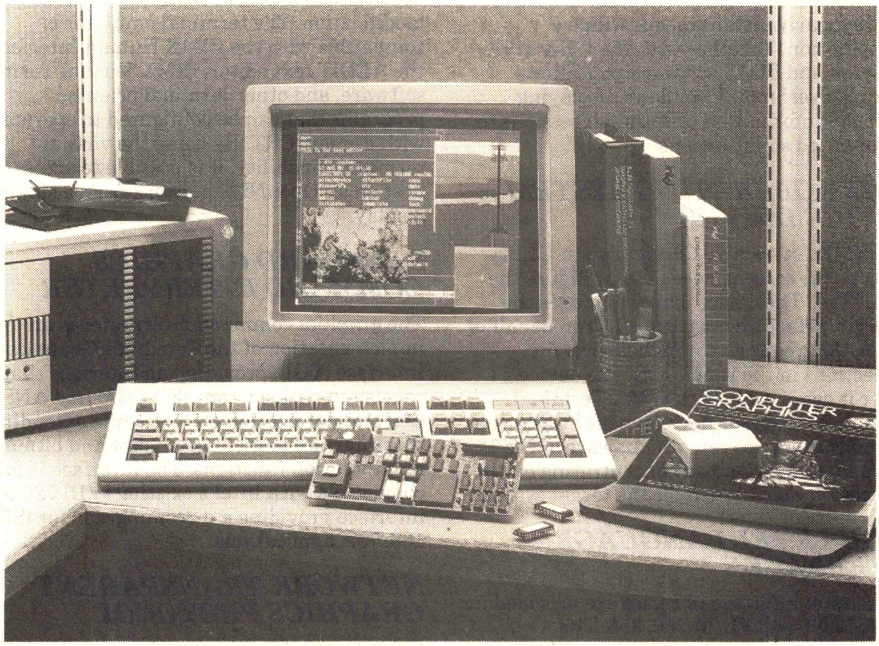
The dual-port memory control block is controlled by PALs and logic on the Intel iSBC 386/116 board. Any combination of chips (microcontrollers, PALs, etc.) could be used to perform the function of the DPMCB. Also, the DPMCB controls arbitration between dual-port access form the PSB and the local bus.

CHAPTER 7: SUGGESTED READING

1. Intel Corporation "MPC User's Guide", Order Number 176526-001.
2. Intel Corporation "iSBC 386/116 and 386/120 Single Board Computer User's Guide", Order Number 451833-001.
3. Intel Corporation "iSBC 286/100A Single Board Computer User's Guide", Order Number 149093-002.
4. Intel Corporation 'iSBC 186/100 Single Board Computer User's Guide", Order Number 148732-001.
5. Intel Corporation "iAPX 86/88, 186/188 User's Manual", Order Number 210912-001.
6. Intel Corporation, AP-423, "The MULTIBUS II Interconnect Design Guide", Order Number 280640-001.
7. *MULTIBUS II: The Message Passing Coprocessor (MPC) and System Design,* Mini/Micro Northeast: Professional Program Session Record 28, 1987.
8. Intel Corporation "80386 Hardware Reference Guide", Order Number 231732-002.
9. Intel Corporation "Embedded Controller Handbook Vol. 1", Order Number 210918-006.

iSBX Expansion Modules 6

iSBX 279A DISPLAY SUBSYSTEM



280667-1

HIGH-SPEED GRAPHICS/WINDOWING FOR iRMX® II SYSTEMS

The iSBX 279A is a complete graphics subsystem designed to provide users of Intel's iRMX® II, and RMX III real-time systems with advanced interactive graphics functions. Based on Intel's 82786 Display Processor, the iSBX 279A efficiently off-loads bitmap and window manipulation from the application CPU, preserving real-time system performance. Windowing, ASCII terminal emulation with system console support, and drawing commands are provided in a convenient system expansion package. Complete software support, including iRMX Device Drivers, Application Interface Libraries for C-286, 386 and PL/M 286, 386 and loadable fonts, provides a high-level, network transparent interface, allowing application portability across Intel's real-time platforms and shortening application development time.

FEATURES

- Intel 82786 Display Processor
- iRMX Operating System Device Drivers
- High-Level Language Interface
- System Console Support Kits
- Windowed User Interface
- Terminal Emulator
- Standard Graphics Command Interpreter
- Network Transparent Graphics Protocol

August 1993
Order Number: 280667-001

intel

FEATURES

INTEL 82786 DISPLAY PROCESSOR

The Intel 82786 Display Processor is used to provide nearly instantaneous window manipulation. Together with the iSBX 279A firmware and iRMX software, this allows multiple overlapping windows displaying graphical information or terminal sessions to be presented simultaneously.

iRMX® OPERATING SYSTEM DEVICE DRIVER

The iRMX Interactive Configuration Utility provides the screens needed to configure the iSBX 279A. The device driver manages the device interface and performs I/O on behalf of application requests through device/file connections in the iRMX IOS. The device driver is compatible with iRMX Terminal Support. This speeds development by allowing the programmer to remain unaware of the device interface, instead concentrating on the application code needed for the target system.

HIGH-LEVEL LANGUAGE INTERFACE

Application interface libraries are provided for C-286, C-386 and PL/M 286, 386. The application interface is a set of graphics and windowing primitives that provide standard drawing functions with complete control of bitmaps and windows. This allows the application programmer to quickly begin writing sophisticated real-time graphics applications using a portable interface for iRMX systems.

WINDOWED USER INTERFACE

The user may interactively MOVE, RESIZE, PUSH, POP, and SELECT windows using the mouse and an easy-to-use menu provided for this purpose. Several terminal sessions and interactive graphics applications can be managed from a single console. Window and bitmap manipulation is performed locally by the iSBX 279A, allowing complex user-interface operations to proceed in parallel with time-critical real-time tasks on the host-CPU.

TERMINAL EMULATOR

The terminal emulator allows existing applications to run in a window without modification. The terminal emulator is compatible with the iRMX Human Interface, the AEDIT text editor, iRMX Virtual Terminal software, and other terminal oriented programs that can be configured to operate with a smart CRT. By using the virtual terminal capability, it is possible to access any host on an OpenNET network from a single display.

STANDARD GRAPHICS COMMAND INTERPRETER

The graphics command interpreter is an implementation of the Computer Graphics Interface (CGI), providing an interface that is consistent with current ISO-CGM and ANSI-CGI standardization efforts, while extending this interface to include window and bitmap manipulation functions. The interface is fully compatible with Intel's existing iVDI 720 R1.8 interface providing a direct upgrade for iSBC 186/78A applications.

NETWORK TRANSPARENT GRAPHICS PROTOCOL

Using OpenNET it is possible to display images stored on remote nodes, run interactive applications from any node on the network, allow multiple applications on several processors to share a single display, and access other displays from a local application processor. Network transparency allows distributed applications to be controlled from a single console.

INTEL QUALITY AND RELIABILITY

The components of the iSBX 279A subsystem are designed and manufactured in accordance with Intel's high quality standards. Quality is assured through rigorous testing in our state-of-the-art Environmental Testing Laboratory.

REAL-TIME GRAPHICS INTERFACE COMMAND SUMMARY

Graphics Server Commands			
gqbeginmetafile	gqendmetafile	gqbeginpicture	gqendpicture
gqsetbackgroundcolor	gqsetvdcextent	gqsetcliprectangle	gqsetclipindicator
gqclearviewsurface	gqresetdefaults	gqsetdeviceviewport	gqopencgi
gqclosecgi	gqsetscalingmode	gqpolyline	gqline
gqpolymarker	gqmarker	gqpolygon	gqrectangle
gqcircle	gqarc	gqarcclose	gqcellarray
gqtext	gqappendtext	gqarccenter	gqarccenterclose
gqsetasfs	gqsetpolylinebundleindex	gqsetpolylinebundlerep	gqsetlinetype
gqsetlinewidth	gqsetlinecolor	gqsetpolymarkerbundleindex	gqsetpolymarkerbundlerep
gqsetmarkertype	gqsetmarkersize	gqsetmarkercolor	gqsetfillareabundleindex
gqsetfillareabundlerep	gqsetinteriorstyle	gqsetfillcolor	gqsethatchindex
gqsetpatternindex	gqsetpatterntable	gqsetpatternreferencepoint	gqsetpatternsize
gqsetperimetertype	gqsetperimeterwidth	gqsetperimetercolor	gqsettextfontindex
gqsettextfontlist	gqsettextbundleindex	gqsettextbundlerep	gqsettextprecision
gqsetcharacterexpansionfactor	gqsetcharacterspacing	gqsettextcolor	gqsetcharacterheight
gqsetcharacterorientation	gqsetcharacterpath	gqsettextalignment	gqsetcolortable
gqsettransparency	gqsetdisplaycorner	gqsetblinkmode	gqsetblinkcolor
gqsetblinkinterval	gqsetimagebounds	gqdisplayimage	gqsetcursorcolor
gqsetwritemask	gqsetcursorsize	gqcreatewindow	gqdeletewindow
gqdeleteallwindows	gqcreatebitmap	gqdeletebitmap	gqdeleteallbitmaps
gqsetwindowpriority	gqpopwindow	gqpushwindow	gqchangebitmap
gqselectdrawingbitmap	gqmovewindow	gqpanbitmap	gqpanwindow
gqresizewindow	gqsetwindowvisi	gqsetbordervisi	gqsetfieldcolor
gqsetbordercolor	gqset1bppcolorpad	gqset2bppcolorpad	gqset4bppcolorpad
gqsetsyscursorcolor	gqsetcursorvisi	gqloadfont	gqdeletefont
gqsetdrawingmode	gqbitblt	gqloadbitmap	gqloadcursor
gqdeletecursor	gqassigncursor	gqloadmenu	gqdeletemenu
gqselectmenu	gqenablesystemmenu	gqdisablesystemmenu	gqinqerrorinfo
gqinqwindowinfo	gqreqwindowinfo	gqinqbitmapinfo	gqinqcursorinfo
gqinqfontinfo	gqinitializelocator	gqinitializestring	gqreleaseinputdevice
gqsetinputdevicemode	gqsamplelocator	gqsamplestring	gqrequestlocator
gqrequeststring	gqsetpromptstate	gqsetechostate	gqstartinput
gqendinput	gqrequestinput		

Terminal Server Commands		
LINE FEED	BACKSPACE	RETURN
CURSOR UP	CURSOR RIGHT	CURSOR HOME
MOVE CURSOR	INSERT LINE	DELETE LINE
CLEAR SCREEN	CLEAR TO END OF SCREEN	CLEAR TO END OF LINE
VISUAL BELL	DISABLE VISUAL BELL	ENABLE VISUAL BELL
INVERT SCREEN	SET BACKGROUND COLOR	SET FOREGROUND COLOR
RESET TO DEFAULTS	SELECT PRIMARY CHARACTER SET	SELECT SECONDARY CHARACTER SET
SELECT PRIMARY FONT INDEX	SELECT SECONDARY FONT INDEX	USE PRIMARY FONT
USE SECONDARY FONT	START REVERSE VIDEO	END REVERSE VIDEO

Buffered Operating System Interface		
cgi__initialize	cgi__terminate	cgi__send__cmd
cgi__send	cgi__sendreply	cgi__flush

intel

SPECIFICATIONS

- 640 by 480 Pixel Display, 25MHZ Pixel Clock
- 60 HZ Non-interlaced Frame Rate, 31.5Khz Horizontal Rate
- 1 Megabyte of Bitmap Memory
- Bitmap Depths of 1, 2, 4, or 8 bits
- Up to 256 Displayed Colors from a Palette of 16.7 Million[1]
- On-board Self-Test and Diagnostics

Interfaces Supported
IBM[2] PC/AT[3] Keyboard (or compatible)
PC-MOUSE (or compatible) locator device
NEC MultiSync[4] Analog-RGB Monitor (or compatible)
8-bit iSBX Bus Interface (all signals TTL compatible)

Physical Characteristics			
Width:	7.5 in.	≅19.05 cm.	
Length:	3.15 in.	≅8.00 cm.	
Height:	0.80"	≅2.03 cm.	iSBX 279 Only
	1.14 in.	≅2.89 cm.	With iSBC Host
Weight:	6.0 oz.	≅170.1 gm.	iSBX 279 Only

Electrical Requirements
+5 VDC ±5% @≈3.0 A
+12 VDC ±5% @≈0.025 A
−12 VDC ±5% @≈0.025 A

Environmental Requirements	
Operating Temperature:	0°C to 55°C@200 LFPM minimum air flow
Storage Temperature:	−40°C to +70°C
Humidity:	0% to 95%, non-condensing

[1] See the RGI279 Product Release Notes for restrictions on window size/placement when displaying 256 colors.
[2] IBM is a registered trademark of International Business Machines, Inc.
[3] PC/AT is a trademark of International Business Machines, Inc.
[4] MultiSync is a trademark of NEC.

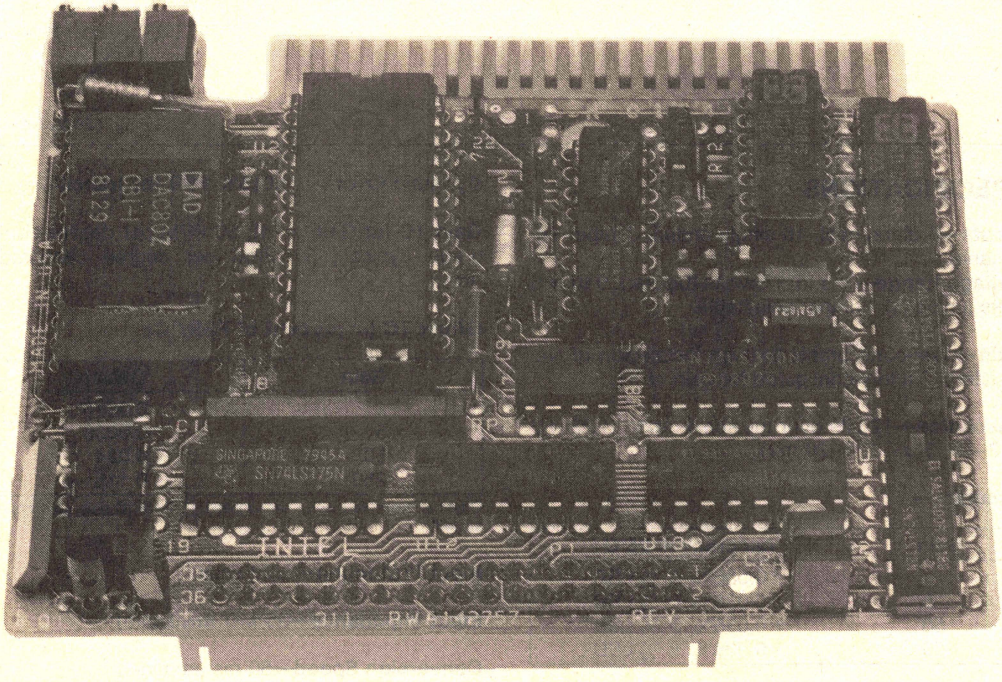
ORDERING INFORMATION

Order Code	*Description*
SBX279A	Display Subsystem (Board Only)

iSBX 311
ANALOG INPUT MULTIMODULE BOARD

- **Low Cost Analog Input Via iSBX Bus Connector**
- **8 Differential/16 Single-Ended, Fault Protected Inputs**
- **20 mV to 5V Full Scale Input Range, Resistor Gain Selectable**
- **Unipolar (0 to +5V) or Bipolar (−5V to +5V) Input, Jumper Selectable**
- **12-Bit Resolution Analog-To-Digital Converter**
- **18 KHz Samples Per Second Throughput to Memory**

The Intel iSBX 311 Analog Input MULTIMODULE board provides simple interfacing of non-isolated analog signals to any iSBC board having an IEEE 959 iSBX I/O Expansion Bus connector. The single-wide iSBX 311 plugs directly onto the iSBC board, providing data acquisition of analog signals from eight differential or sixteen single-ended voltage inputs, jumper selectable. Resistor gain selection is provided for both low level (20 mV full scale range) and high level (5 volt FSR) signals. The iSBX 311 MULTIMODULE board provides 12 bit resolution, 11 bit accuracy, and a simple programming interface, all on a low cost SBX board.

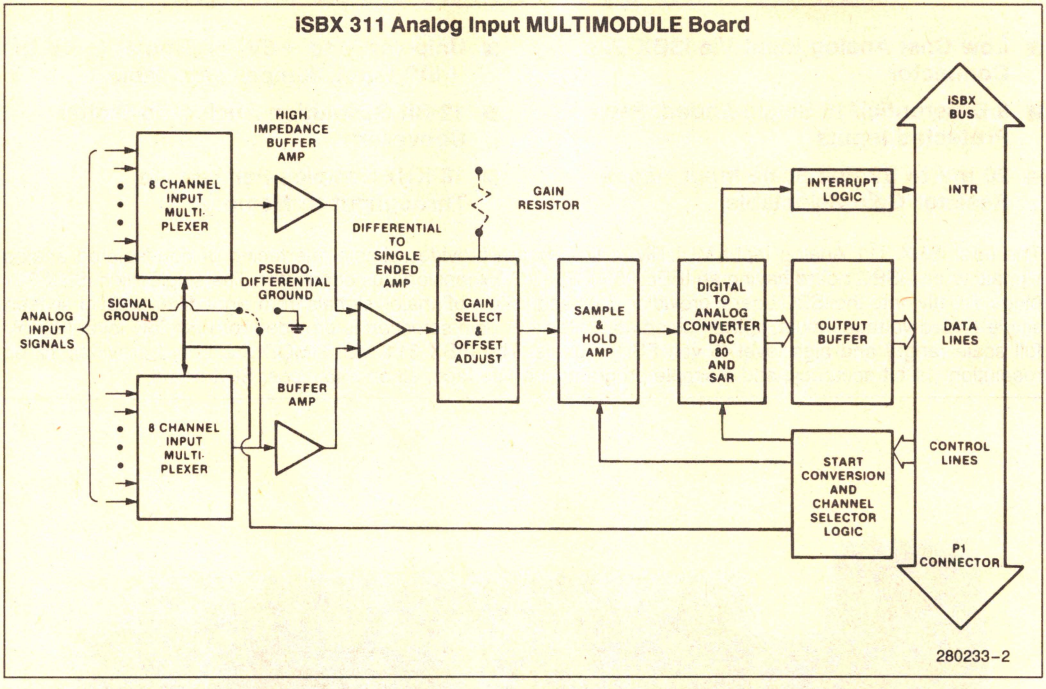


280233-1

6

August 1993
Order Number: 280233-003

BLOCK DIAGRAM



SPECIFICATIONS

Inputs—8 differential. 16 single-ended. Jumper selectable.
Voltage Range—−5 to +5 volts (bipolar). 0 to +5 volts (unipolar). Jumper selectable.

Gain—User-configurable through installation of two resistors. Factory-configured for gain of X1.

Resolution—12 bits over full scale range (1.22 mV at 0–5V, 5 μV at 0–20 mV).

Accuracy—

Gain	Accuracy at 25°C
1	±0.035% ± ½ LSB
5	±0.035% ± ½ LSB
50	±0.035% ± ½ LSB
250	±0.035% ± ½ LSB

NOTE:
Figures are in percent of full scale reading. At any fixed temperature between 0° and 60°C, the accuracy is adjustable to ±0.035% of full scale.

Dynamic Error—±0.015% FSR for transitions.

Gain TC (at Gain = 1): 30 PPM per degree centigrade (typical); 56 PPM per degree centigrade (max).

Offset TC (in percent of FSR/°C):

Gain	Offset
1	0.0018
5	0.0036
50	0.024
250	0.116

Offset is measured with user-supplied 10 PPM/°C gain resistors installed.

Input Protection—±30 volts.

Input Impedance—20 MΩ (minimum).

Conversion Speed—50 ms (nominal).

Common Mode Rejection Ratio—60 db (minimum).

Sample and hold—sample time 15 ms.

Aperture—hold aperture time: 120 ns.

Connectors—

Interface	Pins (Qty)	Centers		Mating Connectors
		in	cm	
J1 8/16 Channels Analog	50	0.1	0.254	3m 3415-000

Physical Characteristics

Width: 9.40 cm (3.7 inches)

Length: 6.35 cm (2.5 inches)

Height: 2.03 cm (0.80 inch) MULTIMODULE board only

2.82 cm (1.13 inches) MULTIMODULE and iSBC board

Weight: 68.05 gm (2.4 ounces)

Electrical Characteristics (from iSBX connector)

V_{cc} = ±5 volts (±0.25V), I_{cc} = 250 mAmax
V_{dd} = +12 volts (±0.6V), I_{dd} = 50 mAmax
V_{ss} = −12 volts (±0.6V), I_{ss} = 55 mAmax

Environmental Characteristics

Operating Temperature: 0° to 60°C (32° to 140°C)

Relative Humidity: to 90% (without condensation)

Reference Manuals

142913— iSBX 311 Analog Input MULTIMODULE Board Hardware Reference Manual (order separately)

ORDERING INFORMATION

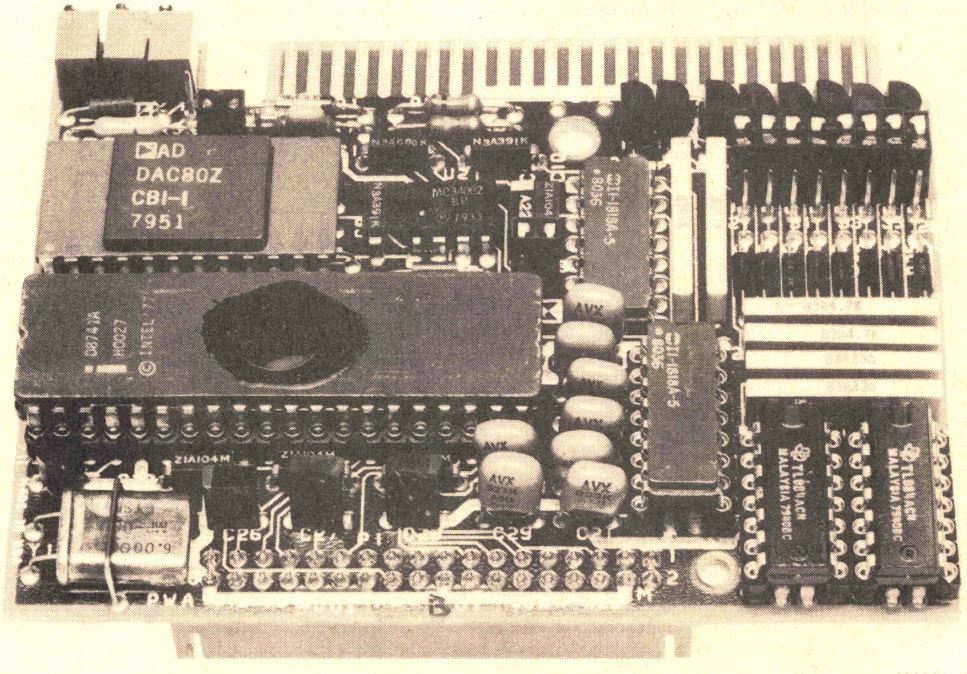
Order Code	Description
SBX311	Analog Input MULTIMODULE Board

6

iSBX 328
ANALOG OUTPUT MULTIMODULE BOARD

- **Low Cost Analog Output Via iSBX Bus Connector**
- **8 Channel Output, Current Loop or Voltage in any Mix**
- **4–20 mA Current Loop; 5V Unipolar or Bipolar Voltage Output**
- **12-Bit Resolution**
- **0.035% Full Scale Voltage Accuracy @ 25°C**
- **Programmable Offset Adjust in Current Loop Mode**

The Intel iSBX 328 MULTIMODULE board provides analog signal output for any intelligent board having an IEEE 959 iSBX I/O Expansion bus connector. The single-wide iSBX 328 plugs directly onto the host board, providing eight independent output channels of analog voltage for meters, programmable power supplies, etc. Voltage output can be mixed with current loop output for control of popular 4–20 mA industrial control elements. By using an Intel single chip computer (8041) for refreshing separate sample-hold amplifiers through a single 12 bit DAC, eight channels are contained on a single SBX board for high density and low cost per channel. High quality analog components provide 12 bit resolution, and slew rates per channel of 0.1V per microsecond. Maximum channel update rates are 5 KHz on a single channel to 1 KHz on all eight channels.

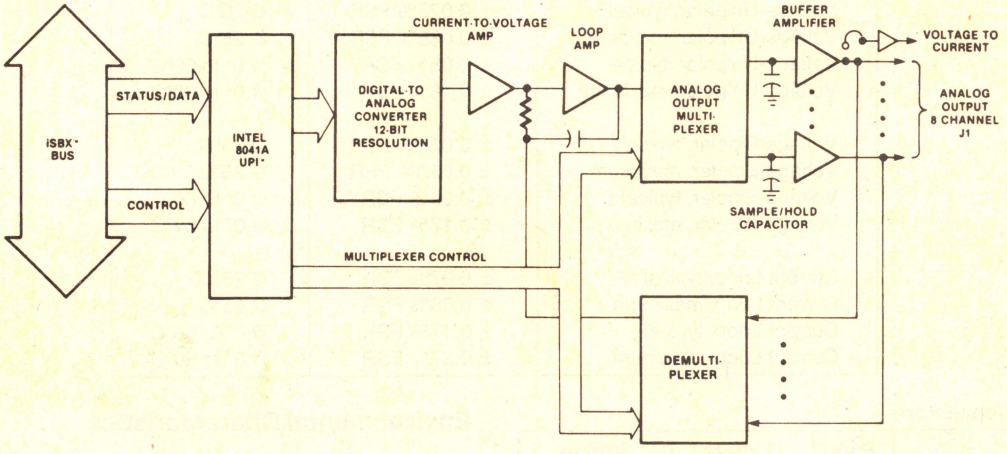


280234–1

August 1993
Order Number: 280234-002

BLOCK DIAGRAM

iSBC 328 Analog Output MULTIMODULE Board Block Diagram



280234-2

SPECIFICATIONS

Outputs — 8 non-isolated channels, each independently jumpered for voltage output or current loop output mode.

Voltage Ranges — 0 to +5V (unipolar operation) −5 to +5V (bipolar operation)

Current Loop Range — 4 to 20 mA (unipolar operation only)

Output Current — ±5 mA maximum (voltage mode-bipolar operation)

Load Resistance — 0 to 250Ω with on-board iSBX power. 1000Ω minimum with 30 VDC max. external supply

Compliance Voltage — 12V using on-board iSBX power. If supplied by user, up to 30 VDC max

Resolution — 12 bits bipolar or unipolar

Slew Rate — 0.1V per microsecond minimum

Single Channel Update Rate — 5 KHz

Eight Channel Update Rate — 1 KHz

Output Impedance — 0.1Ω. Drives capacitive loads up to 0.05 microfarads. (approx. 1000 foot cable)

Temperature Coefficient — 0.005%/°C

Refresh and Throughput Rates**

Refresh 1 channel (no new data):	80 μs
Refresh all 8 channels (no new data):	650 μs
Update and refresh 1 channel with new data: firmware program 2	150 μs
for each additional channel	130 μs
Update and refresh 1 channel with new data: firmware program 1 or 3	200 μs
for each additional channel	155 μs
Update and refresh all 8 channels (all new data): firmware program 2	1,050 ms
per channel of new data	50 μs
Update and refresh all 8 channels (all new data): firmware program 1or 3	1,280 ms
per channel of new data	80 μs

**All times nominal

6

Accuracy—

Mode	Accuracy	Ambient Temp
Voltage-Unipolar, typical	± 0.025% FSR	@ 25°C
Voltage-Unipolar, maximum	± 0.035% FSR	@ 25°C
Voltage-Unipolar, typical	± 0.08% FSR	@ 0° to 60°C
Voltage-Unipolar, maximum	± 0.19% FSR	@ 0° to 60°C
Voltage-Bipolar, typical	± 0.025% FSR	@ 25°C
Voltage-Bipolar, maximum	± 0.035% FSR	@ 25°C
Voltage-Bipolar, typical	± 0.09% FSR	@ 0° to 60°C
Voltage-Bipolar, maximum	± 0.17% FSR	@ 0° to 60°C
Current Loop, typical	± 0.07% FSR	@ 25°C
Current Loop, maximum	± 0.08% FSR	@ 25°C
Current Loop, typical	± 0.17% FSR	@ 0° to 60°C
Current Loop, maximum	± 0.37% FSR	@ 0° to 60°C

Connectors—

Interface	Pins (Qty)	Centers in	Centers cm	Mating Connectors
J1 8/16 channels analog	50	0.1	0.254	3m 3415-000

Physical Characteristics

Width: 9.40 cm (3.7 inches)

Length: 6.35 cm (2.5 inches)

Height: 1.4 cm (0.56 inch) MULTIMODULE board only

2.82 cm (1.13 inches) MULTIMODULE and iSBC board.

Weight: 85.06 gm (3.0 ounces)

Electrical Characteristics

V_{CC} = ±5V (0.25V), I_{CC} = 140 mA max

V_{DD} = ±12V (±0.6V), I_{DD} = 45 mA max (voltage mode)

= 200 mA max (current loop mode

V_{SS} = −12V (±0.6V), I_{SS} = 55 mA max

Environmental Characteristics

Operating Temperature: 0° to 60°C (32° to 140°C)

Relative Humidity: to 90% (without condensation)

Reference Manuals

142914— iSBX 328 Analog Output MULTI-MODULE Board Hardware Reference Manual (Order Separately)

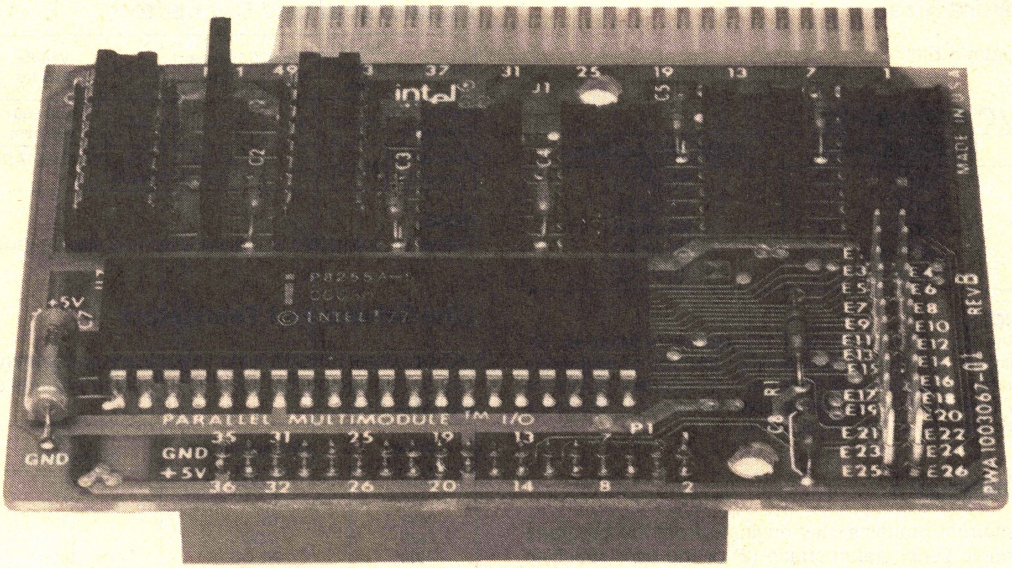
ORDERING INFORMATION

Order Code	Description
SBX328	Analog Output MULTIMODULE Board

iSBX 350
PARALLEL I/O MULTIMODULE BOARD

- **IEEE 959 SBX Bus Compatible I/O Expansion**
- **24 Programmable I/O Lines with Sockets for Interchangeable Line Drivers and Terminators**
- **Three Jumper Selectable Interrupt Request Sources**
- **Accessed as I/O Port Locations**
- **Single +5V Low Power Requirement**
- **SBX Bus On-Board Expansion Eliminates Multibus System Bus Latency and Increases System Throughput**

The Intel iSBX 350 Parallel I/O MULTIMODULE Board is a member of Intel's line of IEEE 959 SBX I/O Expansion Bus products. The iSBX 350 module provides 24 programmable I/O lines with sockets for interchangeable line drivers and terminators. The board is closely coupled to the host board through the SBX bus, and as such, offers maximum on-board performance and frees Multibus system traffic for other system resources. In addition, incremental power dissipation is minimal at only 1.6 watts (not including optional driver/terminators).



280235-1

6

SPECIFICATIONS

I/O Capacity

24 programmable lines

Parallel I/O Port Operation Modes

Port	Lines (qty)	Mode of Operation					Control
		Unidirectional				Bidirectional	
		Input		Output			
		Unlatched	Latched & Strobed	Latched	Latched & Strobed		
A	8	X	X	X	X	X	
B	8	X	X	X	X		
C	4	X		X			X(1)
	4	X		X			X(1)

NOTE:
1. Part of Port C must be used as a control port when either Port A or Port B are used as a latched and strobed input or a latched and strobed output port or Port A is used as a bidirectional port.

Word Size

Data: 8 Bits

I/O Addressing

82C55A-5 Ports	iSBX 350 Address
Port A	X0 or X4
Port B	X1 or X5
Port C	X2 or X6
Control	X3 or X7
Reserved	X8 to XF

NOTE:
The first digit of each port I/O address is listed as "X" since it will change dependent on the type of host single board computer used. Refer to the Hardware Reference Manual for your host single board computer to determine the first digit of the port address.

Interrupts

Interrupt requests may originate from the programmable peripheral interface (2) or the user specified I/O (1).

Transfer Rates

Maximum rates are 2.8 MBytes/sec.

NOTE:
Actual transfer speed is dependent upon the cycle time of the host microcomputer.

Parallel Interface Connectors

Interface	No. of Pairs/ Pins	Centers (in.)	Connector Type	Vendor	Vendor Part No.
Parallel I/O Connector	25/50	0.1	Female	3M	3415-0001 with Ears
Parallel I/O Connector	25/50	0.1	Female Soldered	GTE Sylvania	6AD01251A1DD

NOTE:
Connector compatible with those listed may also be used.

Line Drivers and Teminators

I/O Drivers—The following line drivers and terminators are all compatible with the I/O driver sockets on the iSBX 350.

Driver	Characteristic	Sink Current (mA)
7438	I, OC	48
7437	I	48
7432	NI	16
7426	I, OC	16
7409	NI, OC	16
7408	NI	16
7403	I, OC	16
7400	I	16

NOTE:
I = Inverting, NI = Non-Inverting, OC = Open Collector

Port 1 has 25 mA totem pole drivers and 1 kΩ terminators.

I/O Terminators—220Ω/330Ω divider or 1 kΩ pull up - user supplied.

Physical Characteristics

Width: 7.24 cm (2.85 in.)

Length: 9.40 cm (3.70 in.)

Height*: 2.04 cm (0.80 in.) iSBX 350 Board

2.86 cm (1.13 in.) iSBX 350 Board + Host Board

Weight: 51 gm (1.79 oz)

*See Figure 2

Electrical Characteristics

DC Power Requirements

Power Requirements	Configuration
+5 @ 320 mA	Sockets XU3, XU4, XU5, and XU6 empty (as shipped).
+5V @ 500 mA	Sockets XU3, XU4, XU5, and XU6 contain 7438 buffers.
+5V @ 620 mA	Sockets XU3, XU4, XU5, and XU6 contain iSBC 901 termination devices.

Environmental

Operating Temperature: 0°C to +55°C

Reference Manual

9803191—iSBX 350 Parallel I/O MULTIMODULE Manual (NOT SUPPLIED)

ORDERING INFORMATION

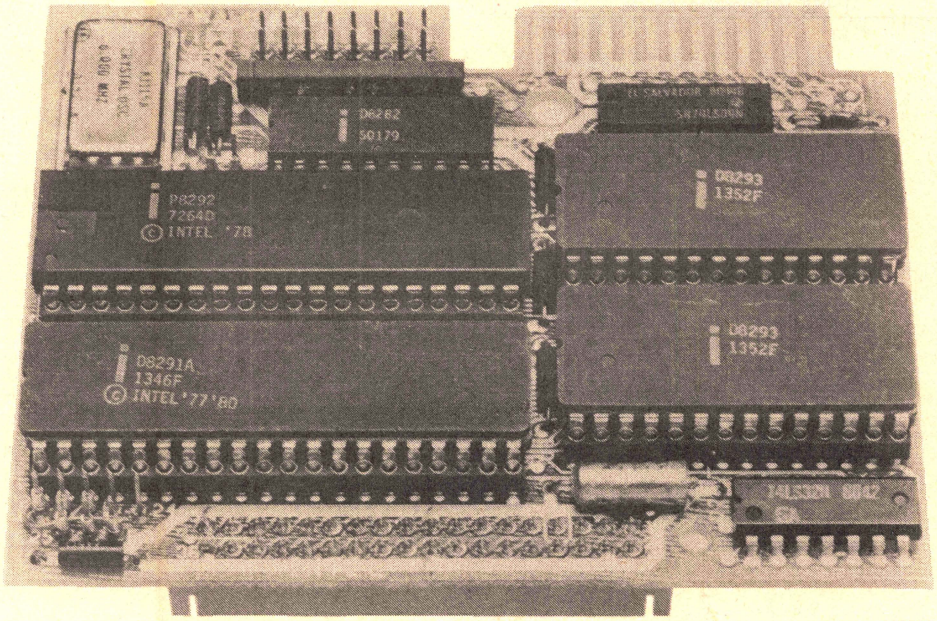
Order Code	Description
pSBX350	Parallel I/O MULTIMODULE Board

iSBX 488
GPIB MULTIMODULE BOARD

- **Complete IEEE 488-1978 Talker/ Listener Functions Including:**
  - **— Addressing, Handshake Protocol, Service Request, Serial and Parallel Polling Schemes**
- **Complete IEEE 488-1978 Controller Functions Including:**
  - **— Transfer Control, Service Requests and Remote Enable**
- **Simple Read/Write Programming**
- **Software Functions Built into VLSI Hardware for High Performance, Low Cost and Small Size**
- **Standard iSBX Bus Interface for Easy Connection to Intel iSBC Boards**
- **IEEE 488-1978 Standard Electrical Interface Transceivers**
- **Five Volt Only Operation**

The Intel iSBX 488 GPIB Talker/Listener/Controller MULTIMODULE board provides a standard interface from any Intel iSBC board equipped with an IEEE 959 iSBX I/O Expansion Bus connector to over 600 instruments and computer peripherals that use the IEEE 488-1978 General Purpose Interface Bus. By taking full advantage of Intel's VLSI technology the single-wide iSBX 488 MULTIMODULE board implements the complete IEEE 488-1978 Standard Digital Interface for Programmable Instrumentation on a single low cost board. The iSBX 488 MULTIMODULE board includes the 8291A GPIB Talker/Listener, 8292 GPIB Controller and two 8293 GPIB Transceiver devices. This board represents a significant step forward in joining microcomputers and instrumentation using industry standards such as the MULTIBUS system bus, iSBX bus and IEEE 488-1978. The high performance iSBX 488 MULTIMODULE board mounts easily on Intel iSBX bus compatible single board computers.

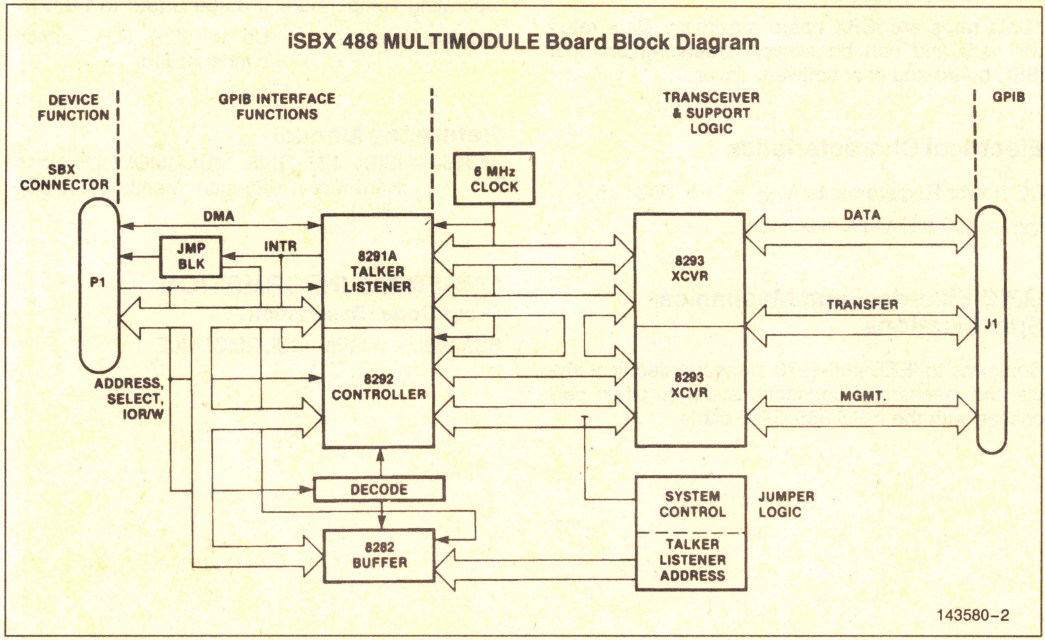
A simple user programming interface for easy reading, writing and monitoring of all GPIB functions is provided. This intelligent interface minimizes the impact on host processor bandwidth.



143580-1

September 1993
Order Number: 143580-002

BLOCK DIAGRAM



Installation

The iSBX 488 MULTIMODULE board plugs directly onto the female iSBX connector available on Intel iSBC boards. The MULTIMODULE board is then secured at one additional point with nylon hardware (supplied) to insure the mechanical security of the assembly.

SPECIFICATIONS

Physical Characteristics

Width: 3.70 in (0.94 cm)

Length: 2.85 in (7.24 cm)

Height: 0.8 in (2.04 cm)

Weight: 3.1 oz (87.8 gm)

GPIB Data Rate*

300K bytes/sec transfer rate with DMA host iSBC board

GPIB Functions Supported

IEEE 488-1978 Functions(1)

Function	ISBX 488 Supported IEEE Subsets
Source Handshake (SH)	SH0, SH1
Acceptor Handshake (AH)	AH0, AH1
Talker (T)	T0 through T8
Extended Talker (TE)	TE0 through TE8
Listener (L)	L0 through L4
Extended Listener (LE)	LE0 through LE9
Service Request (SR)	SR0, SR1
Remote Local (RL)	RL0, RL1
Parallel Poll (PP)	PP0, PP1, PP2
Device Clear (DC)	DC0 through DC2
Device Trigger (DT)	DT0, DT1
Controller (C)	C0 through C28

NOTE:
1. For detailed information refer to IEEE Standard Digital Interface for Programmable Instrumentation published by The Institute of Electrical and Electronics Engineers, Inc. 1978.

6

50K bytes/s transfer rate using programmed I/O
730 ns Data Accept Time

*Data rates are iSBX board maximum. Data rates will vary and can be slower depending on host iSBC board and user software driver.

Electrical Characteristics

DC Power Requirements: V_{CC} = +5 VDC ±5%

I_{CC} = 600 milliamps maximum

GPIB Electrical and Mechanical Specifications

Conforms to IEEE 488-1978 standard electrical levels and mechanical connector standard when purchased with the iSBC 988 GPIB cable.

Environmental Characteristics

Operating Temperature: 0° to 60°C (32° to 140°F)

Relative Humidity: Up to 90% R.H. without condensation.

Reference Manual

143154— iSBX 488 GPIB MULTIMODULE Board Hardware Reference Manual (not supplied).

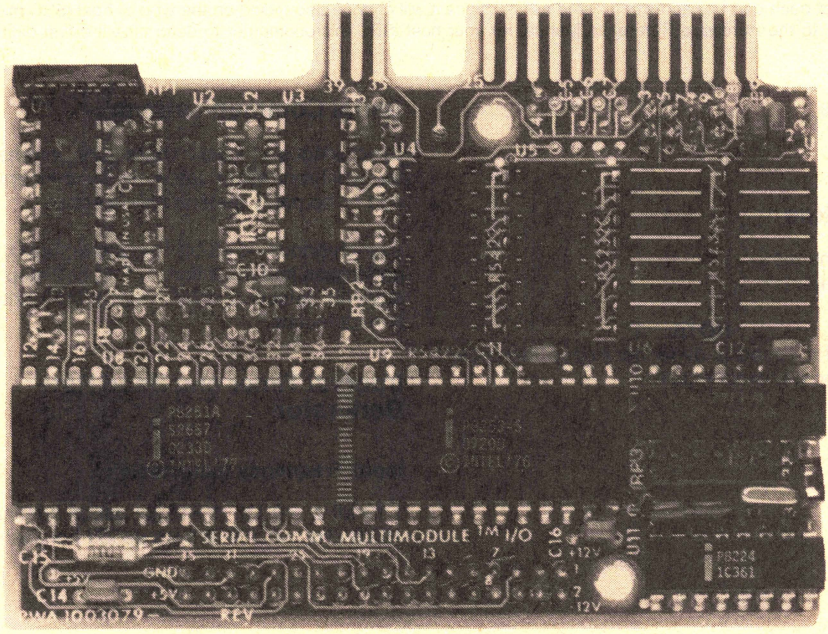
ORDERING INFORMATION

Order Code	Description
SBX488	GPIB MULTIMODULE

iSBX 351
SERIAL I/O MULTIMODULE BOARD

- **IEEE959 iSBX Bus Compatible I/O Expansion**
- **Programmable Synchronous/ Asynchronous Communications Channel with RS232C or RS449/422 Interface**
- **Software Programmable Baud Rate Generator**
- **Two Programmable 16-Bit BCD or Binary Timer/Event Counters**
- **Four Jumper Selectable Interrupt Request Sources**
- **Accessed as I/O Port Locations**
- **Low Power Requirements**
- **Single +5V when Configured for RS449/422 Interface**
- **iSBX Bus On-Board Expansion Eliminates MULTIBUS System Bus Latency and Increases System Throughput**

The Intel iSBX 351 Serial I/O MULTIMODULE board is a member of Intel's line of IEEE 959 iSBX I/O Expansion Bus compatible products. The iSBX 351 module provides one RS232C or RS449/422 programmable synchronous/asynchronous communications channel with software selectable baud rates. Two general purpose programmable 16-bit BCD or binary timers/event counters are available to the host board to generate accurate time intervals under software control. The iSBX board is closely coupled to the host board through the SBX bus, and as such, offers maximum on-board performance and frees MULTIBUS system traffic for other system resources. In addition, incremental power dissipation is minimal requiring only 3.0 watts (assumes RS232C interface).



280236-1

6

SPECIFICATIONS

I/O Addressing

I/O Address for an 8-Bit Host	I/O Address for a 16-Bit Host	Chip Select	Function
X0, X2, X4 or X6	Y0, Y4, Y8 or YC	8251A USART	Write: Data Read: Data
X1, X3, X5 or X7	Y2, Y6, YA or YE	MCS0/ Activated (True)	Write: Mode or Command Read: Status
X8 or XC	Z0 or Z8	8253 PIT	Write: Counter 0 Load: Count (N) Read: Counter 0
X9 or XD	Z2 or ZA	MSC1/Activated (True)	Write: Counter 1 Load: Count N Read: Counter 1
XA or XE	Z4 or ZC		Write: Counter 2 Load: Count (N) Read: Counter 2
XB or XF	Z6 or ZE		Write: Control Read: None

NOTE:
X = The iSBX base address that activates MCS0 & MSC1 for an 8-bit host.
Y = The iSBX base address that activates MCS0 for a 16-bit host.
Z = The iSBX base address that activates MCS1 for a 16-bit host.
The first digit, X, Y or Z, is always a variable, since it will depend on the type of host microcomputer used. Refer to the Hardware Reference Manual for your host microcomputer to determine the first digit of the I/O base address.
The first digit of each port I/O address is listed as "X" since it will change depending on the type of host iSBC microcomputer used. Refer to the Hardware Reference Manual for your host iSBC microcomputer to determine the first digit of the I/O address.

Word Size

Data—8 bits

Access Time

Read—250 ns max
Write—300 ns max

NOTE:
Actual transfer speed is dependent upon the cycle time of the host microcomputer.

Serial Communications

Synchronous—5–8-bit characters; internal character synchronization; automatic sync insertion; even, odd or no parity generation/detection.

Asynchronous—5–8-bit characters; break character generation and detection; 1, 1½, or 2 stop bits; false start bit detection; even, odd or no parity generation/detection.

Interval Timer and Baud Rate Generator

Input Frequency (selectable):

1.23 MHz ±0.1% (0.813 μs period nominal)
153.6 kHz ±0.1% (6.5 μs period nominal)

Sample Baud Rate

8253 PIT[1] Frequency (kHZ, Software Selectable)	8251 USART Baud Rate (Hz)[2]		
	Synchronous	Asynchronous	
		÷16	÷64
307.2	—	19200	4800
153.6	—	9600	2400
76.8	—	4800	1200
38.4	38400	2400	600
19.2	19200	1200	300
9.6	9600	600	150
4.8	4800	300	75
2.4	2400	150	—
1.76	1760	110	—

NOTES:
1. Frequency selected by I/O writes of appropriate 16-bit frequency factor to Baud Rate Register.
2. Baud rates shown here are only a sample subset of possible software-programmable rates available. Any frequency from 18.75 Hz to 614.4 kHz may be generated utilizing on-board crystal oscillator and 16-bit Programmable Interval Timer (used here as frequency divider).

Output Frequency

	Rate Generator (Frequency)		Real-Time Interrupt (Interval)	
	Min	Max	Min	Max
Single Timer[1]	18.75 Hz	614.4 kHz	1.63 μs	53.3 ms
Single Timer[2]	2.34 Hz	76.8 kHz	13.0 μs	426.7 ms
Dual Timer[3] (Counters 0 and 1 in Series)	0.000286 Hz	307.2 kHz	3.26 μs	58.25 min
Dual Timer[4] (Counters 0 and 1 in Series)	0.0000358 Hz	38.4 kHz	26.0 μs	7.77 hrs

NOTES:
1. Assuming 1.23 MHz clock input.
2. Assuming 153.6 kHz clock input.
3. Assuming Counter 0 has 1.23 MHz clock input.
4. Assuming Counter 0 has 153.6 kHz clock input.

Interrupts

Interrupt requests may originate from the USART (2) or the programmable timer (2).

Interfaces

IEEE959 iSBX I/O Expansion Bus

Serial—configurable of EIA Standards RS232C or RS449/422

EIA Standard RS232C signals provided and supported.

Clear to Send (CTS)
Data Set Ready (DSR)
Data Terminal Ready (DTR)
Request to Send (RTS)
Receive Clock (RXC)
Receive Data (RXD)
Transmit Clock (DTE TXC)
Transmit Data (TXD)

EIA Standard RS449/422 signals provided and supported.

Clear to Send (CS)
Data Mode (DM)
Terminal Ready (TR)
Request to Send (RS)
Receive Timing (RT)
Receive Data (RD)
Terminal Timing (TT)
Send Data (SD)

Physical Characteristics

Width: 7.24 cm (2.85 inches)

Length: 9.40 cm (3.70 inches)

Height*: 2.04 cm (0.80 inches)
iSBX 351 Board
2.86 cm (1.13 inches)
iSBX 351 Board and Host Board

Weight: 51 grams (1.79 ounces)

Electrical Characteristics

DC Power Requirements

Mode	Voltage	Amps (Max)
RS232C	+5V ±0.25V	460 mA
	+12V ±0.6V	30 mA
	−12V ±0.6V	30 mA
RS449/422	+5V ±0.25V	530 mA

Environmental Characteristics

Temperature: 0°C–55°C, free moving air across the base board and MULTIMODULE board.

Reference Manual

9803190— iSBX 351 Serial I/O MULTIMODULE Manual (NOT SUPPLIED)

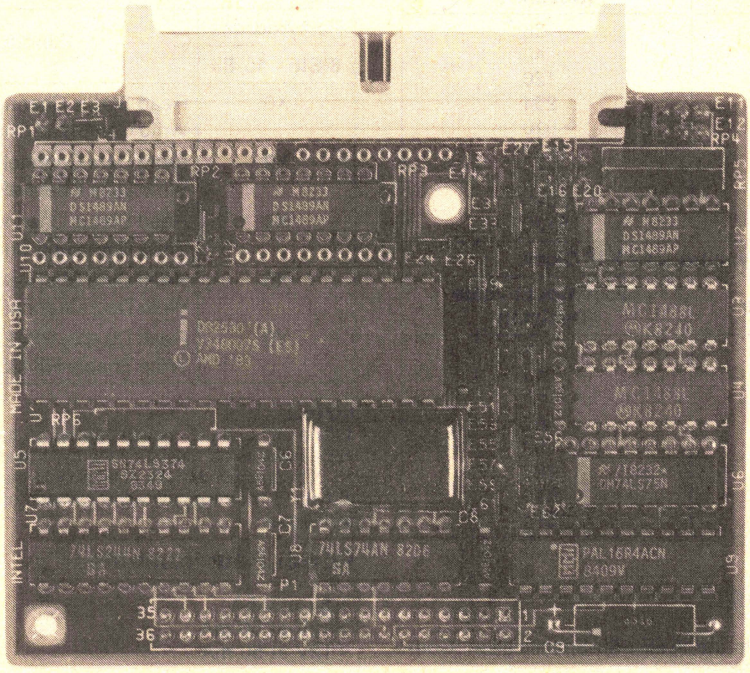
ORDERING INFORMATION

Order Code	Description
SBX351	Serial I/O MULTIMODULE Board

iSBX 354 DUAL CHANNEL SERIAL I/O MULTIMODULE BOARD

- **Two RS232C or RS422A/449 Programmable Synchronous/ Asynchronous Communications Channels**
- **Programmable Baud Rate Generation for Each Channel**
- **Full Duplex Operation**
- **IEEE 959 SBX Bus Compatible I/O Expansion**
- **Supports HDLC/SDLC, NRZ, NRZI or FM Encoding/Decoding**
- **Three Interrupt Options for Each Channel**
- **Low Power Requirements**

The Intel iSBX 354 Serial I/O MULTIMODULE board is a member of Intel's line of IEEE 959 iSBX I/O Expansion Bus products. Utilizing Intel's 82530 Serial Communications Controller component, the iSBX 354 module provides two RS232C or RS422A/449 programmable synchronous/asynchronous communications channels. The 82530 component provides two independent full duplex serial channels, on chip crystal oscillator, baud-rate generator and digital phase locked loop capability for each channel. The iSBX board connects to the host board through the iSBX bus. This offers maximum on-board performance and frees the MULTIBUS System bus for use by other system resources.



280045-1

6

SPECIFICATIONS

Word Size

Data—8 bits

Clock Frequency

4.9152 MHz

Serial Communications

Synchronous—Internal or external character synchronization on one or two synchronous characters

Asynchronous—5–8 bits and 1, 1½ or 2 stop bits per character; programmable clock factor; break detection and generation; parity, overrun, and framing error detection

Sample Baud Rate:

Synchronous X1 Clock	
Baud Rate	**82530 Count Value (Decimal)**
64000	36
48000	49
19200	126
9600	254
4800	510
2400	1022
1800	1363
1200	2046
300	8190
Asynchronous X16 Clock	
Baud Rate	**82530 Count Value (Decimal)**
19200	6
9600	14
4800	30
2400	62
1800	83
1200	126
300	510
110	1394

INTERFACES

SBX Bus: Meets the IEEE 959 Specification, Compliance Level: D8 F

Serial: Meets the EIA RS232C standard on Channels A and B. Meets the EIA RS422A/449 standard on Channels A and B, Multi-drop capability on Channel A only.

Signals Provided

RS232C DTE

-Transmit Data
-Receive Data
-Request to Send
-Clear to Send
-Data Set Ready
-Signal Ground
-Carrier Detect
-Transmit Clock (2)
-Receive Clock
-Data Terminal Ready
-Ring Indicator

RS232C DCE

-Transmit Data
-Receive Data
-Clear to Send
-Data Set Ready
-Signal Ground
-Carrier Detect
-Transmit Clock (2)
-Receive Clock
-Ring Indicator

RS422A/449

-Send Data
-Receive Timing
-Receive Data
-Terminal Timing
-Receive Common

I/O Port Addresses

Port Address		Function
8-Bit	16-Bit	
X0		Read Status Channel B Write Command Channel B
X2		Read Data Channel B Write Data Channel B
X4		Read Status Channel A Write Command Channel A
X6		Read Data Channel A Write Data Channel A
Y0		Read Disable RS422A/449 Buffer Write Enable RS422A/449 Buffer

NOTES:
1. The "X" and "Y" values depend on the address of the iSBX interface as viewed by the base board.
2. "X" corresponds with Activation of the MCS0/interface signal; "Y" corresponds with Activation of the MCS1/interface signal.

Power Requirements

+5V at 0.5A
+12V at 50 mA
−12V at 50 mA

Physical Characteristics

Width: 2.85 inches
Length: 3.70 inches
Height: 0.8 inches
Weight: 85 grams

ENVIRONMENTAL CHARACTERISTICS

Temperature: 0°C to 55°C operating at 200 linear feet per minute across baseboard and MULTIMODULE board

Humidity: To 90%, without condensation

ORDERING INFORMATION

Order Code	Description
SBX354	Dual Channel I/O MULTIMODULE

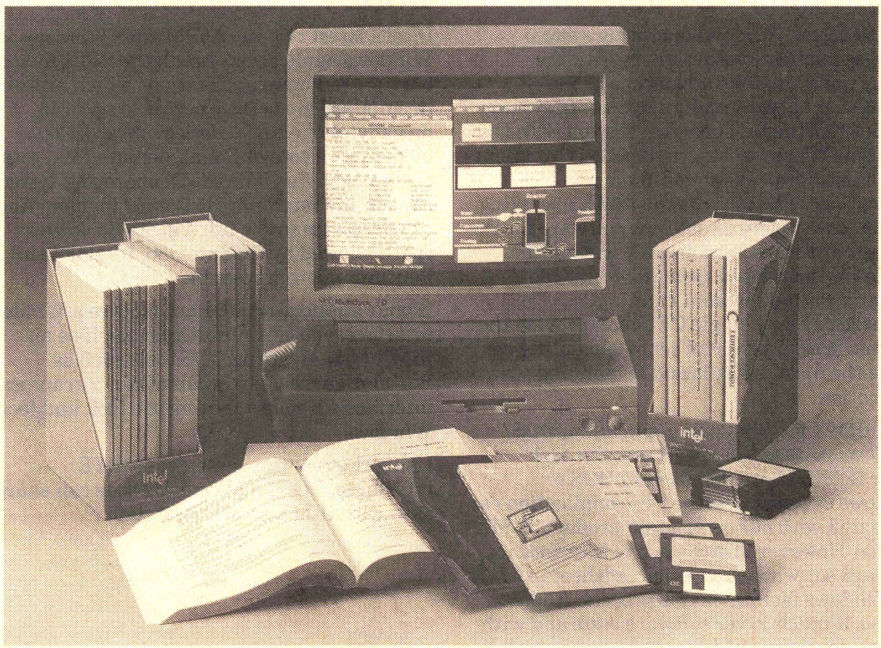
REFERENCE MANUAL

146531—iSBX 354 Channel Serial I/O Board Hardware Reference Manual

Real-Time Software

7

iRMX® FOR WINDOWS* OPERATING SYSTEM



281116-1

REAL-TIME ON A PC

iRMX® for Windows has the ability to run real-time applications on a PC, resulting in significant hardware cost savings over traditional dedicated real-time computers. The PC has become the computer of choice for real-time control because of its versatility, low cost, and availability. The iRMX for Windows operating system runs on many popular PCs* based on the Intel386™, Intel486™ and Pentium™ microprocessors, making it the perfect companion for PC-based control. Designers can enjoy the security of knowing that their development work is compatible with the industry-standard hardware that is driving future technology.

If a desktop PC is not ideal for your application, support for a range of industrialized PC-compatible hardware platforms is also available from Intel and other suppliers. Configurations for the following are included in the product:

Multibus I	Multibus II
iSBC 386SX OX Family	iSBC 486 SX 2500 iSBC 486 DX 3300 iSBC 486 DX 6600

7

iRMX® is a registered trademark and Intel386™ and Intel486™ are trademarks of Intel Corporation.
*All other names and brands are the property of their respective owners.
*You should test your choice of software and hardware. Please call for a free iRMX® software demonstration kit to start your evaluation.

October 1993
Order Number: 281116-003

REAL-TIME ON A PC

DOS AND WINDOWS RUN CONCURRENTLY WITH iRMX

The iRMX for Windows operating system provides applications designers with the unique ability to take advantage of the capabilities of two standard operating systems. One is DOS, the PC operating system with the Microsoft Windows 3.x graphical environment and thousands of existing software applications. The other is the iRMX operating system, the standard real-time software for the Intel architecture, used in over one million microprocessor-based systems worldwide. These operating systems, running concurrently on a single CPU, can provide the designer with a powerful set of capabilities to tackle the most difficult and demanding application.

In an iRMX for Windows application, DOS/Windows runs as the lowest priority task under iRMX. If you do not need DOS or Windows system calls, then the application can be created with our companion product, iRMX for PCs. However, if you want to interface to standard software such as spreadsheets, iRMX for Windows facilitates this process. And it does so at much lower software costs and with shorter development times than were possible in the past.

OPERATION WITH STANDARD MODE WINDOWS AND DOS

Intel's latest release of iRMX for Windows lets Windows applications break the 640K memory barrier. Windows applications which are run in standard mode have access to up to 15 megabytes of extended memory, with 1 megabyte of conventional memory for a total of 16 megabytes. The iRMX operating system runs in the remaining extended memory and it can access up to the full 4 gigabyte address space allowed by the Intel386, Intel486 and Pentium microprocessors.

If your application does not require a Windows interface, you can run DOS by itself as an iRMX task. By using DOS tools such as Borland Turbo C*, you can attain a graphical interface without imposing the look and feel of Windows.

Using the real-time extensions (RTE) interfaces, DOS and Windows code can share data with iRMX code in real time.

DYNAMIC DATA EXCHANGE (DDE)

Applications using the Dynamic Data Exchange Interface have the ability to continuously pass data back and forth between themselves and the real-time iRMX tasks. For example, Microsoft Word for Windows* documents or cells of Excel* spreadsheets can be tied directly to iRMX tasks and can be continuously updated with information or can be used to define control points within the real-time control process.

In addition, the iRMX DDE router allows DDE communications between DDE-capable applications throughout your network, letting you create powerful distributed applications.

NETWORK CONNECTIVITY

INCREASED NETWORK CONNECTIVITY

Intel's iRMX for PCs operating system delivers a rich set of networking capabilities, enabling developers to design their applications to run on today's standard networks.

iRMX-NET is part of Intel's family of OpenNET Local Area Network (LAN) products. iRMX-NET enables iRMX computers to interconnect and operate easily with other OpenNET servers and clients running a variety of operating systems.

Each node on iRMX-NET can be used as a server and a client concurrently. This provides great flexibility and saves the cost of a dedicated server.

iRMX computers can also be configured to run standard Message Interprocess Protocol (MIP) or iTP4 jobs for an ISO OSI transport and datalink programmatic interface. These jobs do not require the full iRMX-NET.

Intel has also provided a UNIX-compatible environment on iRMX for PCs to host the TCP/IP* package giving the developer even more networking options. The components of TCP/IP can be divided into two major categories: (1) the kernel, which provides the protocol management, and (2) the applications, which are the human interfaces to TCP/IP.

The TCP/IP kernel contains a Transport Layer (TCP and UDP), Network Layer (IP), and Datalink Layer. On the iRMX operating system, the TCP/IP kernel is a single object module of shared code that is sysloaded. All kernel configuration options are in a single file, easily modified either with a text editor such as AEDIT or with the provided administration commands.

The user's programmatic access to TCP/IP kernel functions is provided through a Berkeley-compatible socket interface library. Tasks communicating across a network use sockets as end-points of a two-way communication path. Tasks create and bind sockets to a network address. The user can specify either stream sockets (supported by TCP) or datagram sockets (supported by UDP).

The primary applications provided are the FTP (File Transfer Protocol) client, which allows file exchanges between remote and local computers and the TELNET client, which allows virtual terminal login sessions to remote computers.

The iRMX for Windows operating system runs standard MS-DOS/Windows as a task, enabling applications to benefit from compatibility with most MS-DOS networking software. However, since networking is an inherently real-time application, we strongly recommend using the networking software (iRMX-Net, TCP/IP) provided with the operating system. Also, because the iRMX operating system is multitasking, it is possible to provide both client and server functionality simultaneously, something impossible in an MS-DOS environment.

DESIGNED FOR RELIABILITY AND EASE OF DEVELOPMENT

Real-time applications are easier to develop with real-time software. Operating systems designed for general business use (like DOS and UNIX*) typically lack essential real-time features, causing development to be expensive, difficult or even impossible. In contrast, iRMX for Windows is designed to make the development of real-time applications easy and successful.

The iRMX for Windows operating system includes a rich set of real-time programming facilities:

- Pre-emptive, priority-based scheduling
- Support for multiple tasks
- Inter-task communication
- Deterministic program execution
- Control of critical resources through regions

In addition, the iRMX for Windows operating system provides a memory protection scheme that detects coding errors. When iRMX for Windows software allocates memory, it assigns a combination read, write, and execute-only status to the allocated code and data segments.

If the application code attempts to execute outside this range or write to a data segment marked read-only, the operating system issues a protection error. User written exception handlers can then gracefully deal with the error condition.

intel

DESIGNED FOR RELIABILITY AND EASE OF DEVELOPMENT

INCREASED FUNCTIONALITY AND PERFORMANCE WITH THE iRMK REAL-TIME KERNEL

The iRMK real-time kernel replaces the real-time kernel of the iRMX nucleus. This provides additional functionality and better performance for iRMX customers as well as an upgrade path for many iRMK customers.

The iRMX user gets additional features such as:

- Alarm handling
- System-wide task creation/deletion/switch handlers
- The ability to notify multiple tasks from an interrupt handler
- Kernel semaphores and mailboxes
- Ability to disable and enable scheduling

SERVICE AND SUPPORT

With the iRMX for Windows operating system, you're not alone when developing your real-time application. Intel has the best technical sales support in the real-time business. The one year software support contract packaged with a development kit includes:

- Large discounts on software version releases
- Free intermediate software product updates
- Technical phone support thru a 1-800 number
- Subscription service and technical information via:
  - — monthly issues of ";Comments" newsletter
  - — quarterly Troubleshooting Guides
  - — Software Problem Report (SPR) service

In addition, a range of consulting, training, and support services is available from Intel's Systems Engineering Services Group. Members of this support group can use their years of relevant experience with iRMX, MS-DOS, and Windows to help you design, develop, or manufacture your real-time system. Intel Systems Engineers have direct access to Intel's source code and design tools, enabling them to quickly and cost-efficiently design and develop customized software for your target systems.

Call your local sales office for more information on available support.

ORDERING INFORMATION

PRODUCT CODE: RMX4WINDKS(1)

iRMX for Windows Operating System Development Kit includes:

- iRMX for Windows operating system modules for all supported platforms(2)
- Utility programs
- Source examples
- iRMX object libraries
- AEDIT text editor
- On-line help utility and manuals
- 32-bit C compiler
- Soft-Scope III debugger
- PL/M-386 compiler
- ASM386 assembler
- Floating Point Libraries
- Linker Package
- One year of software support

PRODUCT CODE: RMX4WINDK(1)

iRMX for Windows Operating System Development Kit (same as RMX4WINDKS but with 90 days of software support).

PRODUCT CODE: NLRMX4WIN(1)

iRMX for Windows Operating System Node License Product is available for multi-system development sites. Requires purchase of one development kit per site.

PRODUCT CODE: RMX4WINMS

iRMX for Windows Development Kit manual set.

Note 1: The iRMX for Windows Development Kit includes the iRMX for PCs Development Kit.

Note 2: Please call for a free copy of the iRMX Demonstration Package to start your evaluation.

intel®

iRMX® FOR PCs OPERATING SYSTEM



281148-1

iRMX® FOR PCs REAL-TIME OPERATING SYSTEM

The 32-bit iRMX® for PCs operating system provides engineers with the ability to design highly reliable, real-time applications for Intel386™, Intel486™ and Pentium™ microprocessor-based PC-AT compatible computers.

Real-time applications are easier to develop with a real-time operating system. Operating systems designed for general business applications typically lack essential real-time features, so real-time development is often expensive, difficult, or even impossible. In contrast, iRMX III software is pre-emptive, priority-based, multitasking, real-time software designed to make the development of real-time applications easy and successful.

The iRMX family of operating systems has become the standard real-time software for the Intel architecture. Only DOS runs on more Intel based systems! Intel's iRMX operating systems are running throughout the world in operating rooms, air traffic control towers, stock exchanges, factory floors, and in many other time-critical environments.

THE iRMX® FOR PCs OPERATING SYSTEM OFFERS:

- Easy real-time development
- Increased functionality and performance
- Industry-standard hardware compatibility
- Easy upgrades from iRMX II
- Unrestricted multitasking and efficient memory usage
- Increased network connectivity
- Ability to use inexpensive Intel386™, Intel486™ and Pentium™ based personal computers

7

iRMX® is a registered trademark of Intel Corporation.
Intel386™, Intel486™, and Pentium™ are trademarks of Intel Corporation.
*Other brands and names are the property of their respective owners.

October 1993
Order Number: 281148-001

intel

FEATURES

EASY DEVELOPMENT FOR RELIABLE APPLICATIONS

Real-time applications are easier to develop with real-time software. The iRMX for PC's operating system is designed to make the development of real-time applications easy and successful. iRMX for PCs includes an extensive set of true real-time programming facilities:

- Pre-emptive, priority-based scheduling
- Support for multiple tasks
- Deterministic program execution
- Inter-task communication through mailboxes and semaphores
- Control of critical resources through regions and semaphores
- Low interrupt and task-switch latency times

Development is further facilitated with a multi-user development environment, loadable drivers/jobs/OS extensions, an improved C compiler, a set of easy-to-use manuals, and Soft-Scope III, a true dynamic debugger for multitasking applications.

In addition, the iRMX for PCs operating system provides a memory protection scheme that detects coding errors. When iRMX for PCs software allocates memory, it assigns a combination of read, write, and execute-only status to allocated code and data segments. If the application code attempts to execute outside this range, to write to a data segment marked read-only, or to write to a code segment, the operating system issues a protection error. User written exception handlers can then gracefully deal with the error condition.

INCREASED FUNCTIONALITY AND PERFORMANCE WITH THE iRMK REAL-TIME KERNEL

The iRMK real-time kernel is the heart of the iRMX nucleus. This provides additional functionality and better performance for iRMX customers as well as an upgrade path for many iRMK customers.

The iRMX user gets additional features such as:

- Alarm handling
- System-wide task creation/deletion/switch handlers
- The ability to notify multiple tasks from an interrupt handler
- Kernel semaphores and mailboxes
- Ability to disable and enable scheduling

INDUSTRY-STANDARD HARDWARE COMPATIBILITY

Applications can be developed more efficiently due to hardware compatibility with the industry standard PC/AT architecture. The iRMX for Windows operating systems supports IBM and Compaq-compatible PCs(1) based on the Intel386, Intel486, and Pentium processors, making it the perfect companion for PC-based control. Designers can enjoy the security of knowing that their development work is compatible with the industry-standard hardware that is driving the future technology. If a desktop PC is not ideal for your application, support for a range of industrialized PC-compatible hardware platforms is also available from Intel and other suppliers. Configurations for the following are included in the product:

NOTE 1: Please call for a free copy of the iRMX for Windows Demonstration Package to verify compatibility with your PC.

Multibus I	Multibus II
iSBC 386SX	iSBC 486SX2500 iSBC 486DX3300 iSBC 486DX6600

FEATURES

EASY iRMX® II UPGRADES

Binary compatibility with 16-bit iRMX II software lets iRMX II users easily move their applications to 32 bits on the iRMX for PCs operating system with the Intel386, Intel486 or Pentium microprocessors. Most 16-bit applications run without change. Selected parts of an application can be moved to 32 bits while others can remain unchanged.

UNRESTRICTED MULTITASKING AND EFFICIENT MEMORY USAGE WITH SHARED C LIBRARIES

The shared C library is implemented as an operating system extension to iRMX for PCs and hence, provides unrestricted multitasking. Instead of each C application having to link to its own copy of the library, there is one system-wide reentrant library which can be shared by all tasks/ jobs within the system. Applications are linked to an interface library which provides access to the shared C library. Since application code need only link to the interface library, significant code size savings are realized.

All floating point routines are provided in the shared C library so you no longer need separate libraries for floating vs. non-floating point applications. The shared C library is linked to the standard floating point libraries.

The Shared C Library is available in two varieties:

1. *The Loadable Shared C Library* is a job which can be loaded via the human interface.
2. *The Resident Shared C Library* is a first-level job which is built into the bootable operating system. It is fully reconfigurable via the ICU.

INCREASED NETWORK CONNECTIVITY

Intel's iRMX for PCs operating system delivers a rich set of networking capabilities, enabling developers to design their applications to run on today's standard networks.

iRMX-NET is part of Intel's family of OpenNET Local Area Network (LAN) products. iRMX-NET enables iRMX computers to interconnect and operate easily with other OpenNET servers and clients running a variety of operating systems.

Each node on iRMX-NET can be used as a server and a client concurrently. This provides great flexibility and saves the cost of a dedicated server.

iRMX computers can also be configured to run standard Message Interprocess Protocol (MIP) or iTP4 jobs for an ISO OSI transport and datalink programmatic interface. These jobs do not require the full iRMX-NET.

Intel has also provided a UNIX-compatible environment on iRMX for PCs to host the TCP/IP* package giving the developer even more networking options. The components of TCP/IP can be divided into two major categories: (1) the kernel, which provides the protocol management, and (2) the applications, which are the human interfaces to TCP/IP.

The TCP/IP kernel contains a Transport Layer (TCP and UDP), Network Layer (IP), and Datalink Layer. On the iRMX operating system, the TCP/IP kernel is a single object module of shared code that is sysloaded. All kernel configuration options are in a single file, easily modified either with a text editor such as AEDIT or with the provided administration commands.

The user's programmatic access to TCP/IP kernel functions is provided through a Berkeley-compatible socket interface library. Tasks communicating across a network use sockets as end-points of a two-way communication path. Tasks create and bind sockets to a network address. The user can specify either stream sockets (supported by TCP) or datagram sockets (supported by UDP).

The primary applications provided are the FTP (File Transfer Protocol) client, which allows file exchanges between remote and local computers. and the TELNET client, which allows virtual terminal login sessions to remote computers.

intel

SERVICE AND SUPPORT

With the iRMX operating system, you are not alone when developing your real-time application. Intel has the best technical support in the real-time business.

The one year software support contract packaged with the development kit includes:

- Large discounts on SW version releases
- Free intermediate SW product updates
- Technical phone support through an 800 number
- Subscription service and technical information via:
  - — monthly issues of "Comments" newsletter
  - — quarterly Troubleshooting Guides
  - — software problem report service

In addition, a range of consulting, training, and support services are available.

ORDERING INFORMATION

PRODUCT CODE: RMX4PCDKS

iRMX for PCs Operating System Development Kit includes:

- Modules for all supported platforms
- Utility programs
- Source examples
- iRMX object libraries
- AEDIT text editor
- On-line help utility and manuals
- 32-bit ANSI standard C compiler
- Soft-Scope III debugger
- PL/M-386 compiler
- ASM386 assembler
- Floating Point Libraries
- Linker Package
- One year of software support

PRODUCT CODE: RMX4PCDK

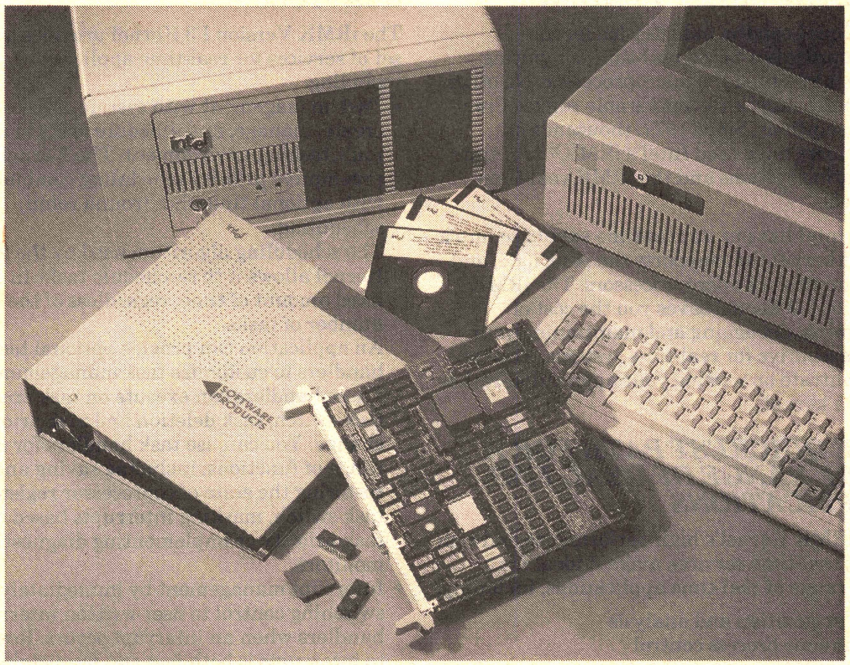
iRMX for PCs Operating System Development Kit (same as above) but with 90 days of software support.

PRODUCT CODE: NLRMX4PC

iRMX for PCs Node License Product is available for multi-system development sites. Requires purchase of one developer kit per site.

intel®

iRMK VERSION I.3 REAL-TIME KERNEL



280613-1

A 32-BIT REAL-TIME KERNEL

The iRMK Version I.3 Real-time Kernel is the 32-bit real-time executive developed, sold and supported by Intel. It reduces the cost and risk of designing and maintaining software for numerous real-time applications such as embedded control systems and dedicated real-time subsystems in multiple-processor systems.

KEY FEATURES

- 32-bit, real-time multitasking kernel
- Rich set of real-time services
- Designed and optimized for the i386™ and i376 microprocessor families
- Comprehensive development tool support
- Extremely fast execution with predictable response times for time-critical applications
- Optional Multibus II multiprocessing support
- Requires only the i386 DX, i386 SX, i486™, or i376 microprocessor; optional support for i387™ and i387 SX numeric coprocessors and other peripheral devices
- Easy customized and enhanced
- Supported by Intel

7

*IBM is a registered trademark of the International Business Machines Corporation.
MS-DOS is a trademark of Microsoft Corporation.
VAX is a registered trademark of Digital Equipment Corporation.
VMS is a trademark of Digital Equipment Corporation.
386/DOS-Extender is a trademark of Phar Lap Software, Inc.
Soft-Scope is a trademark of Concurrent Sciences, Inc.
MetaWare and High C are trademarks of MetaWare, Incorporated.
MicroWay and NDP C-386 are trademarks of MicroWay, Inc.

October 1993
Order Number: 280613-003

intel

FEATURES

REAL-TIME SOFTWARE FROM THE INDUSTRY LEADER

Intel has been the industry leader in microprocessor-based real-time computing since it invented the microprocessor. No other company supplies a comparable range of real-time solutions. Since 1977, thousands of customers have used Intel's iRMX® real-time operating systems and the iRMK real-time kernels.

Now Intel has put its real-time expertise into a 32-bit kernel that supports the i486, i386, and i376 families of microprocessors. The iRMK Version I.3 Kernel saves you the cost of designing, debugging and maintaining your own executive for real-time systems. You can concentrate on writing your application rather than a kernel.

THE QUICKEST PATH FOR A WIDE RANGE OF REAL-TIME APPLICATIONS

The iRMK Kernel's high performance and rich set of real-time services make it ideal for a wide range of real-time applications, including:

Data acquisition and analysis
Continuous process control
Discrete process control
Simulation
Medical instrumentation
Test instrumentation
Image processing
Automated test
Avionics and navigation
Field command control
Energy control
Environmental control
Radio control
Satellite communications
Terminals
Graphics work stations
Robots
Signal processing
Laser printing
Front-end concentrators
Host communications

A RICH SET OF REAL-TIME SERVICES

The iRMK Version I.3 Kernel provides a rich set of services for real-time applications, including:

- Task management with system calls to create, manage, and schedule tasks in a multitasking environment. The Kernel offers preemptive priority scheduling combined with optional time-slice (round robin) scheduling.
  The scheduling algorithm used by the iRMK Kernel allows it to reschedule tasks in a fixed amount of time, regardless of the number of tasks.
  An application can provide optional task handlers to customize task management. These handlers can execute on task creation, task switch, task deletion and task priority change. You can use task handlers for a wide range of functions, including saving and restoring the state of coprocessor registers on task switch, masking interrupts based on task priority or implementing diagnostic monitors.
- Interrupt management by immediately switching control to user-written interrupt handlers when an interrupt occurs. Response to interrupts is both fast and predictable. Most of the Kernel's system calls can be executed directly from interrupt handlers.
- Time management, providing single-shot alarms, repetitive alarms and a real-time clock. Alarms can be reset to implement watchdog, also known as deadman, timers.
- Semaphores, regions, and mailboxes for intertask synchronization and communication. Semaphores are used for intertask signalling and synchronization. Regions are binary semaphores used to ensure mutual exclusion and prevent priority inversion when tasks contend for control of system resources. A task holding a region's unit runs at the priority of the highest priority task waiting for the region's unit.

Mailboxes are queues that can hold any number of messages and are used to exchange data between tasks. Either data or pointers can be sent using mailboxes. The kernel allows mailbox messages to be of any length. High-priority messages may be placed (jammed) at the front of the message queue to ensure that they are received before other messages queued at the mailbox.

To ensure that high-priority tasks are not blocked by lower priority tasks, the kernel allows tasks to queue at semaphores and mailboxes in priority order. The kernel also supports first-in, first-out task queuing.

- Memory pool manager that provides fixed and variable block allocation. Memory can be divided into any number of pools. Multiple memory pools might be created for different speed memories or for allocating different size blocks. Access to a memory pool for fixed-sized allocation is always deterministic. The kernel-supplied memory manager works with flat, segmented and paged addressing. Users can write their own memory managers to provide different memory management policies or to support virtual memory.

SUPPORT FOR MULTIPROCESSING VIA THE MULTIBUS II ARCHITECTURE

The Multibus II architecture is designed to optimize multiple-processor designs. This bus:

- Implements a loosely coupled architecture in which inter-processor interrupts and data are exchanged via messages transmitted as packets over the bus;
- Provides fast bus access;
- Allows inter processor signalling at interrupt speeds from as many as 255 sources;
- Provides data transfer rates of up to 32 megabytes per second;
- Allows multiple communication sessions to occur simultaneously between processors;
- Supports up to 21 CPU boards per chassis with each board providing the processor, memory, and I/O needed for its portion of the application;
- Provides registers—called Interconnect Space—on each board, that can be used for dynamic system configuration.

Two optional modules allow iRMK Kernel applications to make full use of the Multibus II architecture. The first module implements message passing, allowing the application to have direct access to the message-passing hardware or to use Intel's Multibus II transport protocol. The second module implements interconnect space access to support dynamic system configuration.

These modules can be used to implement high performance multiple-processor designs that:

- Break a highly complex real-time application into multiple lower-complexity applications distributed across multiple processors
- Distribute an application that's too CPU-intensive for a single processor between several processors
- Provide redundancy
- Dedicate processors to specific tasks
- Provide inter-operation with any operating system or controller board that uses Intel's Multibus II transport protocol, including the iRMX III, iRMX II, iRMK I, and Intel System V/386 operating systems.

HARDWARE REQUIREMENTS AND SUPPORT

The iRMK Kernel requires only a i386 or i376 family microprocessor and sufficient memory for itself and its application. Its design, however, recognizes that many systems use additional programmable peripheral devices and coprocessors. The Kernel provides optional device managers for the:

- i387 and i387 SX Numeric Coprocessors
- 82380 and 82370 Integrated System Peripherals
- 8254 Programmable Interval Timer
- 8259A Programmable Interrupt Controller
- 82530 Serial Communication Controller

An application can supply managers for other devices and coprocessors in addition to or in replacement of the devices listed above.

The iRMK Kernel was designed to be programmed into PROM or EPROM, making it easy to use in embedded designs.

The iRMK Kernel can be used with any system bus including the Multibus I and Multibus II buses. The optional Multibus II message-passing and Interconnect Space access modules use the Message Passing Coprocessor (MPC). The Kernel provides managers to use the 82380/82370 Integrated System Peripherals or the 82258 Advanced DMA controller with an MPC for message passing.

FEATURES

SUPPORT FOR THE Intel386™ AND INTEL376 ARCHITECTURES

The iRMK Kernel provides 32-bit, protected-mode i486, i386 and i376 CPU operation. By default, the kernel and its application execute in a flat memory space of up to 4 gigabytes in a single privilege level. Applications can add support for any mixture of additional protected mode features including:

- Any model of segmentation
- Memory paging
- Virtual memory
- Multiple privilege levels
- Call and trap gates

These protected mode features can be used to increase the reliability of the application by using the processor's hardware to:

- Protect against attempts to write beyond segment bounds (to catch situations like stack overflow or underflow)
- Allow only privileged or trusted code to access key routines and data
- Isolate bugs to single modules so that the rest of the application and the kernel are not corrupted
- Assign access rights to code and data
- Isolate address spaces

To use these features, the application manipulates the processor's descriptor tables. Because the kernel was implemented specifically to support i486, i386 and i376 microprocessor-based applications, it provides an optional descriptor table manager that simplifies protected mode programming. This manager provides system calls to read and write descriptor table entries, to convert addresses from linear to physical and vice versa, and to get a segment's selector.

A MODULAR ARCHITECTURE FOR EASY CUSTOMIZING

The iRMK Kernel offers maximum flexibility so that it can be customized for each application. Each major function such as mailboxes is implemented as a separate module. The kernel's modules are linked together and supplied individually. Simply link the modules you need for your application. An unused module does not need to be linked in and does not increase the size of the kernel in your application. You can also replace any optional kernel module with one that implements specific features required by your application. For example, you might want to replace the kernel's memory manager with one that supports virtual memory.

Core Modules
— Task Manager
— Interrupt Manager
— Time Manager

Optional Modules
— Mailbox Manager
— Semaphore Manager
— Memory Manager
— Descriptor Manager
— Multibus II Message Passing
— Multibus II Interconnect Space Access

Optional Device Managers
— i387 and i387 SX Numeric Coprocessors
— 8259A PIC
— 8254 PIT
— 82380/82370 Integrated System Peripheral PIT and PIC functions

Figure 1 shows the Kernel's modules.

FEATURES

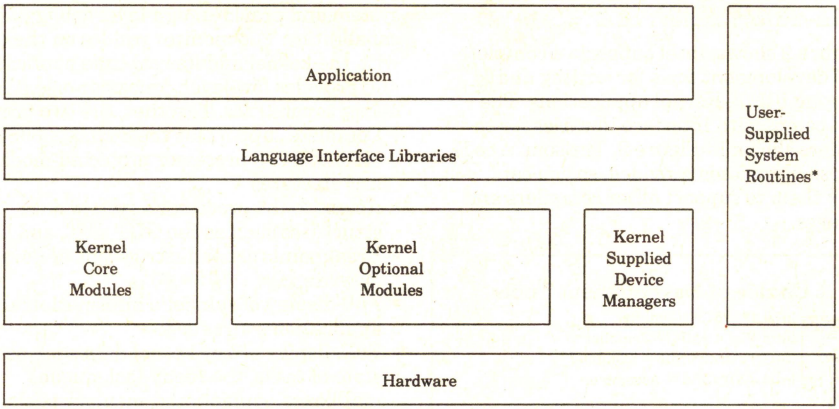


Figure 1: iRMK Real-Time Kernel Architecture
*User-supplied system routines would include interrupt handlers, user-written device managers, and similar routines.

DEVELOPING WITH THE iRMK REAL-TIME KERNEL

As Figure 2 shows, Intel supports a complete line of development tools for writing and debugging iRMK Kernel applications. The kernel comes with interface libraries for the compilers shown in Figure 2. The source code for these libraries is provided, so you can modify them to support other compilers and languages.

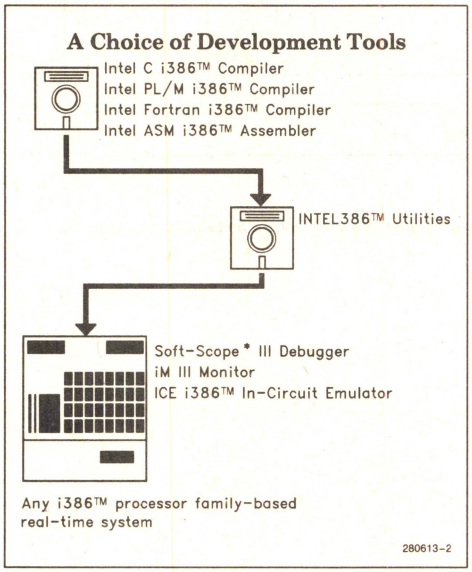


Figure 2: iRMK Kernel Development Tools

THE SOFT-SCOPE III/i386™ AND iM III DEBUGGING TEAM

The Soft-Scope III Debugger and iM III Monitor provide a complete cross-debugging environment for iRMK Kernel applications.

The Debugger resides on a PC/AT-compatible system and provides high-level debugging capabilities. The monitor resides on the target with the kernel and its real-time application and provides low-level, processor-oriented debug capabilities. Together, the two provide:

- Complete support for the 386DX, 386SX, i486, and 376 processor protected-mode architectures
- Access to the processor's system data structures such as the GDT, IDT, and LDT
- Breakpoints set on instructions or data-read or-write
- Full display of symbolic information and the application source code
- Full display of key kernel data such as the state of tasks, the ready-task queue, mailboxes, semaphores, memory pools, ports, and Multibus II message queues and messages
- Powerful macros and C expression evaluation
- 100 kilobytes per minute download between the SoftScope III debugger and the iM III Monitor target

The SoftScope III Debugger runs on PC/AT-compatible systems. It requires a minimum of 512 kilobytes of memory and a serial port. You can configure the iM III Monitor for any target system that contains a 386 family processor, a minimum of 16 kilobytes of memory, and a serial port. The monitor's native command set provides low-level, hardware-oriented debugging capabilities. These basic capabilities can be augmented by extending the monitor itself, using the Soft-Scope III Debugger or writing your own high-level debugger.

FEATURES

CONTENTS OF THE iRMK KERNEL DEVELOPMENT PACKAGE

The iRMK Kernel comes in a comprehensive package that includes:

- Kernel object modules
- Source for the kernel-supplied 82380 and 82370 Integrated System Peripherals; 8259A PIC; 8254 PIT; i387 and i387 SX Numeric Coprocessor; and 82530 Serial Communication Controller device managers.

Source also is provided for the 82258 and 82380 DMA managers for Multibus II message passing.

- Source for language interface libraries
- Source for the call gate interface
- Source for examples showing:
  - — Structure of kernel applications
  - — Use of the kernel with applications written with Intel's PL/M 386, C 386, and FORTRAN 386 compilers and MetaWare's High C 386 compilers
  - — Compile, bind, and build sequences
  - — Sample initialization code for the i386 and i376 microprocessors
  - — Multibus II message passing
  - — Applications written to execute in a flat memory space and in a segmented memory space
- User Reference guide
- 90 days of Customer Support

INTEL SUPPORT, CONSULTING, AND TRAINING

With the iRMK Kernel you get the 386 architecture and real-time software expertise of the Intel's customer support engineers. Intel provides phone support, on- or off-site consulting, troubleshooting guides and updates. The kernel includes 90 days of Intel's Technical Information Phone Service (TIPS). Extended support, consulting and customized training are also available through our field Systems Engineering organization.

SYSTEM CALLS

iRMK VERSION I.3 KERNEL SYSTEM CALLS

KERNEL INITIALIZATION

KN_initialize	Initialize Kernel

OBJECT MANAGEMENT

KN_token_to_ptr	Returns a pointer to the area holding object
KN_current_task	Returns a token for the current task

TASK MANAGEMENT

KN_create_task	Create a task
KN_delete_task	Delete a task
KN_suspend_task	Suspend a task
KN_resume_task	Resume a task
KN_set_priority	Change priority of a task
KN_get_priority	Return priority of a task
KN_stop_scheduling	Suspend task switching
KN_start_scheduling	Resume task switching
KN_set_handler	Set task handler dynamically
KN_reset_handler	Remove previously set task handler

INTERRUPT MANAGEMENT

KN_set_interrupt	Specify interrupt handler

TIME MANAGEMENT

KN_sleep	Put calling task to sleep
KN_create_alarm	Create and start virtual alarm clock

KN_reset_alarm	Reset an existing alarm
KN_delete_alarm	Delete alarm
KN_get_time	Get time
KN_set_time	Set time
KN_tick	Notify Kernel that clock tick has occurred

INTER-TASK COMMUNICATION AND SYNCHRONIZATION

KN_create_semaphore	Create a semaphore
KN_delete_semaphore	Delete a semaphore
KN_send_unit	Add a unit to a semaphore
KN_receive_unit	Receive a unit from a semaphore
KN_create_mailbox	Create a mailbox
KN_delete_mailbox	Delete a mailbox
KN_send_data	Send data to a mailbox
KN_send_priority_data	Place (jam) priority message at head of message queue
KN_receive_data	Request a message from a mailbox

MEMORY MANAGEMENT

KN_create_pool	Create a memory pool
KN_delete_pool	Delete a memory pool
KN_create_area	Create a memory area from a pool
KN_delete_area	Return a memory area to a memory pool
KN_get_pool_attributes	Get a memory pool's attributes

DESCRIPTOR TABLE MANAGEMENT

KN_get_descriptor_attributes	Get a descriptor's attributes
KN_set_descriptor_attributes	Set a descriptor's attributes
KN_initialize_LDT	Initialize local descriptor table (LDT)
KN_null_descriptor	Overwrite a descriptor with the null descriptor
KN_initialize_subsystem	Allows application to be divided into multiple subsystems when application interfaces to Kernel through a call gate
KN_linear_to_ptr	Convert a linear address to a pointer
KN_ptr_to_linear	Convert a pointer to a linear address
KN_get_data_selector	Get the selector for the data segment
KN_get_code_selector	Get the selector for the code segment
KN_translate_ptr	Converts a pointer that will be based on a user-specified selector

82380, 82370, AND 8259A PIC MANAGEMENT

KN_initialize_PICs	Initialize the PICs
KN_mask_slot	Mask out interrupts on a specified slot
KN_unmask_slot	Unmask interrupts on a specified slot
KN_send_EOI	Signal the PIC that the interrupt on a specified slot has been serviced
KN_new_masks	Change interrupt masks
KN_get_slot	Return the most important active interrupt slot

82380, 82370, AND 8254 PIT MANAGEMENT

KN_initialize_PIT	Initialize an 8254 PIT
KN_start_PIT	Start PIT counting
KN_get_PIT_interval	Return PIT interval

i387™ AND i387 SX NUMERIC COPROCESSOR MANAGEMENT

KN_initialize_NDP	Initialize a 387 or 387SX Numeric Coprocessor

CHARACTER I/O DEVICE SUPPORT

KN_initialize_console	Initialize the console device
KN_ci	Read ASCII character from console input device
KN_co	Transfer a character to console output device
KN_csts	Read immediate character, if any, from console input device
KN_initialize_stdio	Initialize stdio library functions
putchar, getchar, printf, scanf	stdio functions for console I/O

MULTIBUS II MESSAGE PASSING MANAGEMENT

KN_initialize_message_passing	Initialize the message passing module
KN_mp_working_storage_size	Compute size of work space needed for message passing
KN_send_tp	Send a transport message
KN_attach_receive_mailbox	Attach a receive mailbox
KN_cancel_tp	Cancel a solicited message or request-response transaction
KN_send_dl	Send a data link message
KN_attach_protocol_handler	Attach a protocol handler
KN_cancel_dl	Cancel a data link buffer request

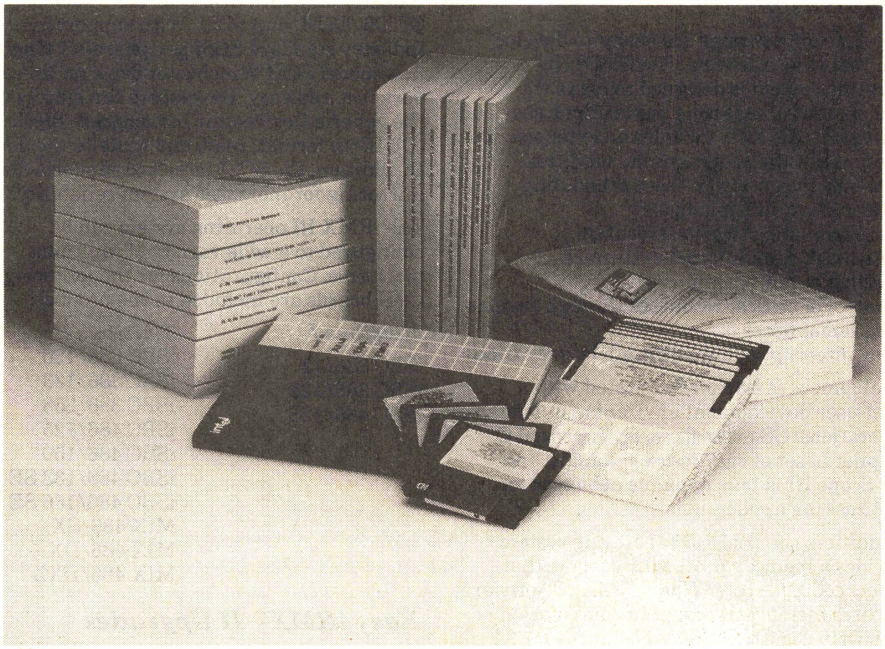
MULTIBUS II INTERCONNECT SPACE MANAGEMENT

KN_initialize_interconnect	Initialize the interconnect module
KN_get_interconnect	Get the value of an interconnect register
KN_set_interconnect	Set the value of an interconnect register
KN_local_host_ID	Get the host ID of the local host

ORDERING INFORMATION

Order Code	Product
RMK RMKIRFX/RMKCRFX	iRMK Version I.3 development software iRMK Kernel incorporation license fees. A separate incorporation license must be signed prior to making incorporations.
IMMIII IMIIIRFX SFTSCPIIID	iM III Monitor development software iM III Monitor incorporation license fees. A separate incorporation license must be signed prior to making incorporations. Soft-Scope III debugger.
SSC-430	Technical Information Support including Phone support, *;Comments* Magazine, and Troubleshooting Guides On- or off-site consultation and customized training on iRMK I.3 are available through the Intel Field Systems Engineering organization. Available on a daily or long term basis.

iRMX® III REAL-TIME OPERATING SYSTEM



281019-1

iRMX® III REAL-TIME OPERATING SYSTEM

The 32-bit iRMX® III operating system provides engineers with the ability to design highly reliable, real-time applications for Intel386™, Intel486™ and Pentium™ microprocessor-based embedded, Multibus I, and Multibus II single board computers.

Real-time applications are easier to develop with a real-time operating system. Operating systems designed for general business applications typically lack essential real-time features, so real-time development is often expensive, difficult, or even impossible. In contrast, iRMX III software is pre-emptive, priority-based, multitasking, real-time software designed to make the development of real-time applications easy and successful.

The iRMX family of operating systems has become the standard real-time software for the Intel architecture. Only DOS runs on more Intel based systems! Intel's iRMX operating systems are running throughout the world in operating rooms, air traffic control towers, stock exchanges, factory floors, and in many other time-critical environments.

The iRMX® III Operating System Offers:

- Easy real-time development
- Increased functionality and performance
- Industry-standard hardware compatibility
- Easy iRMX II upgrades
- Unrestricted multitasking and efficient memory usage
- Increased network connectivity

iRMX® and iSBC are registered trademarks of Intel Corporation.
Intel386™, Intel486™, and Pentium™ are trademarks of Intel Corporation.
*Other brands and names are the property of their respective owners.

October 1993
Order Number: 281019-004

FEATURES

Easy Development for Reliable Applications

Real-time applications are easier to develop with real-time software. The iRMX III operating system is designed to make the development of real-time applications easy and successful. iRMX III includes an extensive set of true real-time programming facilities:

- Pre-emptive, priority-based scheduling
- Support for multiple tasks
- Deterministic program execution
- Inter-task communication through mailboxes and semaphores
- Control of critical resources through regions and semaphores
- Low interrupt and task-switch latency times

Development is further facilitated with a multi-user development environment, loadable drivers/jobs/OS extensions, an improved C compiler, a set of easy-to-use manuals, and Soft-Scope III, a true dynamic debugger for multitasking applications.

In addition, the iRMX III operating system provides a memory protection scheme that detects coding errors. When iRMX III software allocates memory, it assigns a combination of read, write, and execute-only status to allocated code and data segments. If the application code attempts to execute outside this range, to write to a data segment marked read-only, or to write to a code segment, the operating system issues a protection error. User written exception handlers can then gracefully deal with the error condition.

Industry-Standard Hardware Compatibility

Applications can be developed more efficiently due to hardware compatibility with industry standard multi-source peripherals and a wide variety of embedded and Multibus platforms.

The iRMX III operating system supports industry standard disk interfaces such as ST506, ESDI, and SCSI. With support for industry standard SCSI peripherals, bringing a new model SCSI Winchester drive on line is very fast and easy. Developers can simply use the generic Device-unit Information Block (DUIB) to format, read, and write to the device. In addition, the SCSI interface supports disk mirroring for increased system reliability.

The iRMX III operating system can be configured to a wide variety of Multibus platforms. Configurations for the following CPU boards are included in the product:

Multibus I	***Multibus II***
iSBC 386/12	iSBC 386/120
iSBC 386/12S	iSBC 386/133
iSBC 486/12S	iSBC 386/258
	iSBC 486/125
	iSBC 486/150
	iSBC 486/133 SE
	iSBC 486/166 SE
	MIX 486/SX
	MIX 486/DX
	MIX 486/DXZ

Easy iRMX® II Upgrades

Binary compatibility with 16-bit iRMX II software lets iRMX II users easily move their applications to 32 bits on the iRMX III operating system with the Intel386, Intel486 or Pentium microprocessors. Most 16-bit applications run without change. Selected parts of an application can be moved to 32 bits while others can remain unchanged.

Unrestricted Multitasking and Efficient Memory Usage with Shared C Libraries

The shared C library is implemented as an operating system extension to iRMX III and hence, provides unrestricted multitasking. Instead of each C application having to link to its own copy of the library, there is one system-wide library which can be shared by all tasks/jobs within the system. Applications are linked to an interface library which provides access to the shared C library. Since application code need only link to the interface library, significant code size savings are realized.

All floating point routines are provided in the shared C library so you no longer need separate libraries for floating vs. non-floating point applications. The shared C library is linked to the standard floating point libraries.

The Shared C Library is available in two varieties:

1. *The Loadable Shared C Library* is a job which can be loaded via the human interface.
2. *The Resident Shared C Library* is a first-level job which is built into the bootable operating system. It is fully reconfigurable via the ICU.

NETWORK CONNECTIVITY

The iRMX III operating systems (except the iRMX EMB operating system) offer a rich set of networking capabilities, enabling developers to design their applications to run on today's standard networks.

iRMX-NET is part of Intel's family of OpenNET Local Area Network (LAN) products. iRMX-NET enables iRMX-based computers to interconnect and operate easily with other OpenNET servers and clients running a variety of operating systems.

Each node on iRMX-NET can be used as a server and a client concurrently. This provides great flexibility and saves the cost of a dedicated server.

iRMX-based computers can also be programmed to run standard Message Interprocess (MIP) or iTP4 for an ISO OSI transport and datalink programmatic interface. These jobs do not require the full iRMX-NET.

Intel also provides a UNIX-compatible environment to host the TCP/IP package giving the developer even more networking options. The components of TCP/IP can be divided into two major categories:

- the kernel, which provides protocol management
- the applications, which are the human interfaces to TCP/IP

The TCP/IP kernel contains a Transport Layer (TCP and UDP), Network Layer (IP), and Datalink Layer. On the iRMX operating system, the TCP/IP kernel is a single object module of shared code that is loaded at run time. All kernel configuration options are in a single file, easily modified either with a text editor or with the provided administration commands.

The user's programmatic access to TCP/IP kernel functions is provided through a Berkeley-compatible socket interface library. Tasks communicating across a network use sockets as end-points of a two-way communication path. Tasks create and bind sockets to a network address. The user can specify either stream sockets (supported by TCP) or datagram sockets (supported by UDP).

The iRMX III operating systems support the FTP (File Transfer Protocol) client and server, which allows file exchanges between remote and local computers, and TELNET client and server, which allows virtual terminal login sessions to remote computers. Support for NFS (Network File System) is available from a third party software vendor.

Wider Development System Choices

With the introduction of release 2.1 the user will have the ability to develop their iRMX III application using either Microsoft C and C++ or Watcom C and C++ compilers. Interface libraries, header files and class libraries for iRMX objects will be included with the developers kit.

intel

SERVICE AND SUPPORT

With the iRMX III operating systems, you're not alone when developing your real-time application. Intel has the best technical sales support in the real-time business. The one year software support contract packaged with a development kit includes:

- Large discounts on software version releases
- Free intermediate software product updates
- Technical phone support thru a 1-800 number
- Subscription service and technical information via:
  — monthly issues of ";Comments" newsletter
  — quarterly Troubleshooting Guides
  — Software Problem Report (SPR) service

In addition, a range of consulting, training, and support services is available from Intel's Systems Engineering Services Group. Members of this support group can use their years of relevant experience with iRMX, MS-DOS, and Windows to help you design, develop, or manufacture your real-time system. Intel Systems Engineers have direct access to Intel's source code and design tools, enabling them to quickly and cost-efficiently design and develop customized software for your target systems.

Call your local sales office for more information on available support.

ORDERING INFORMATION

Product Code: SYRIIIDKS

iRMX III Operating System Development Kit includes:

- Modules for all supported platforms
- Utility programs
- source examples
- iRMX object libraries
- AEDIT text editor
- On-line help utility and manuals
- 32-bit C compiler
- Soft-Scope III debugger
- PL/M-386 compiler
- ASM386 assembler
- Floating Point Libraries
- Linker Package
- One year of software support

Product Code: SYRIIIDK

iRMX III Operating System Development Kit (same as above) but with 90 days of software support.

Product Code: RMXIIIMNL

iRMX III Operating System Development Kit manual set

Product Code: NLSYRIII

iRMX III Node License Product is available for multi-system development sites. Requires purchase of one developer kit per site.

For further information or to place an order in the U.S. or Canada, **CALL 1-800-GET-iRMX (1-800-438-4769).**

Outside North America, FAX 503-681-8497 or call the appropriate number below:

China (Beijing) Tel: 499-4951

France Tel: 331 30 57 7000

Germany Tel: 49 89 90 99 20

Israel Tel: 972 3 548 3222

Italy Tel: 39 2 892 00950

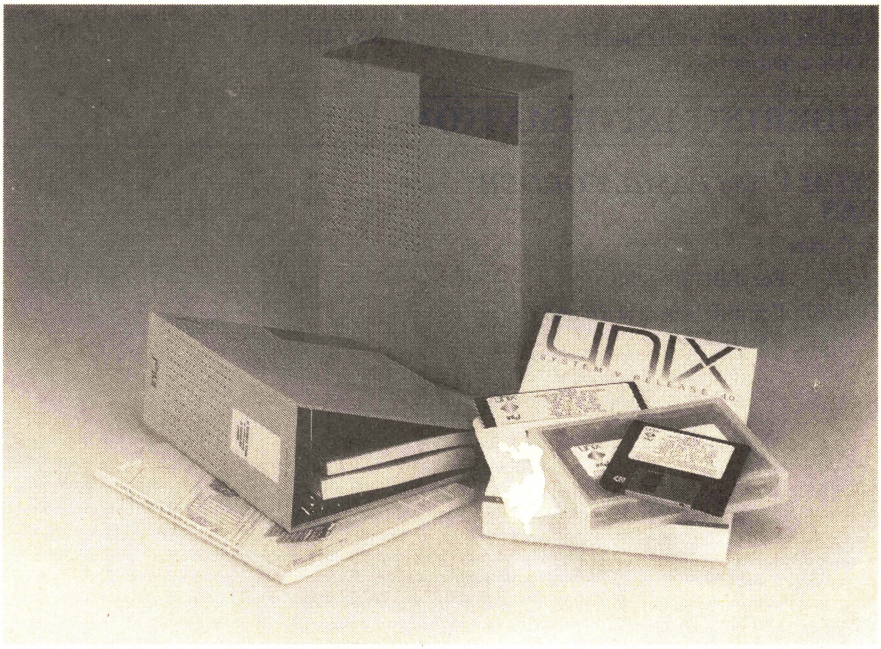
Japan Tel: 0298-47-8511

Netherlands Tel: 311 040 71111

Russia Tel: +7 095 4439785

United Kingdom Tel: +44 793 696000

INTEL SYSTEM V RELEASE 4.03 FOR INTEL'S TRADITIONAL MULTIBUS II CPU BOARDS



280692-1

UNIX SYSTEM V 4.03 FOR MULTIBUS II CPU BOARDS*

Intel's UNIX* System V Release 4.03 for Multibus II delivers the flexibility, integration, and power of UNIX* to Multibus II systems. UNXV4II and UNXV4IISE deliver full support for Intel's Multibus II System Architecture, MSA, Multibus transport, and driver support for Intel's popular Multibus II I/O cards.

UNIX* System V Release 4.03 has the benefits of 20 years of refinement and thousands of applications and tools designed for the UNIX* environment. Networking support is complete with TCP/IP, NFS, and System V OpenNET. Graphical User Interfaces are provided through X Windows and OpenLook*, Motif* is also available, but sold separately.

The UNIX* System V Release 4.03 operating system running on Multibus II is an excellent manager for real-time boards and systems running iRMX or a small real-time kernel such as iRMK. UVXV4II and UNXV4IISE provide message passing, application down loading, standard GUIs, and standard networking. These products communicate with any other OS in the system with Multibus II transport capability.

7

*Other brands and names are the property of their respective owners.

September 1993
Order Number: 280692-003

intel

SYSTEM V/386 OPERATING SYSTEM

FEATURES

- Based on UNIX System V Release 4.03
- Networking support includes TCP/IP, NFS, and SV OpenNet
- X Windows Support with OpenLook (Motif available separately)
- Drivers for Multibus II I/O cards
- Boots directly on iSBC 486/166SE or 486/133SE (UNXV4IISE)
- Can use the iSBC 386/258 as a boot server (UNXV4II)

ORDERING INFORMATION

SYSTEM V/386 FAMILY ORDER CODES

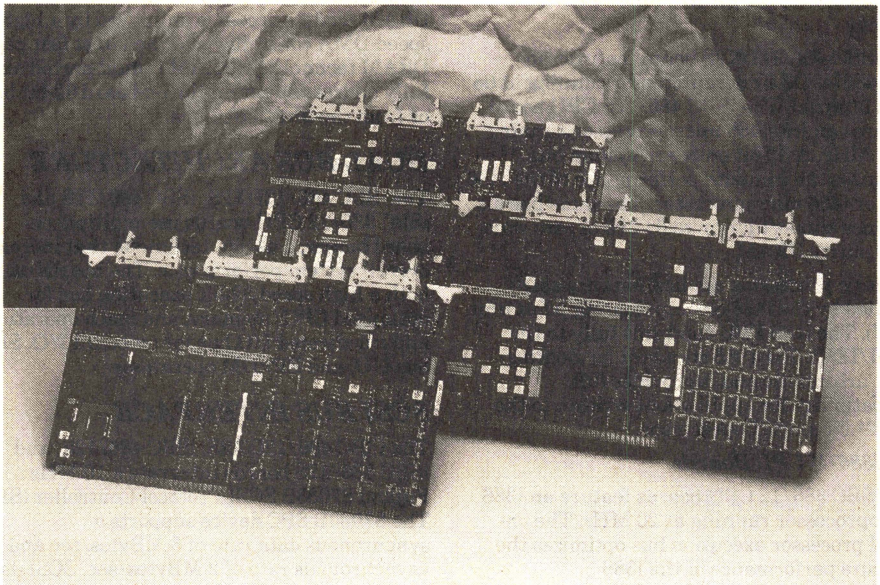
Order Codes

UNXV4II	For iSBC 386/258
UNXV4IISE	For iSBC 486/166SE and 486/133SE

Multibus I Single Board Computers

8

iSBC 486/12DX2S, iSBC 486/12S AND iSBC 386/12 SINGLE BOARD COMPUTERS



281110-1

UPGRADE COMPATIBLE CPU BOARDS BOOT MULTIBUS I PERFORMANCE

The iSBC 486/12DX2S, iSBC 486/12S, and iSBC 386/12 boards are high-performance, high integration Multibus I single board computers. These boards share a common core design that provides functional compatibility with the iSBC 286/12 single board computer while achieving a 2X to 8X performance improvement over a 286 CPU-based board.

The optional SCSI (Small Computer System Interface) is available as an integrated feature on the iSBC 486/12DX2S, iSBC 486/12S, and iSBC 386/12S. SCSI data transfers to local DRAM are DMA supported and FIFO buffered, enabling sustained synchronous SCSI data rates up to 5 MBytes/sec.

FEATURES

- 66 MHz i486™ DX2, 33 MHz i486 or 20 MHz i386™ Microprocessors
- Optional i387™ Numeric Coprocessor on i386 products
- 82258 Advanced DMA Controller
- Dual Bus Architecture—Dedicated 32-bit processor execution bus
- 2-64 MB on-board 32-bit dual-port parity DRAM on i486 boards
- 1-16 MB on-board 32-bit dual-port parity DRAM on i386 boards
- SCSI Interface—5 MBytes/sec synchronous data rate
- Two 32-pin JEDEC EPROM sites
- Two serial ports via 8274
- 82C55A parallel port
- Three on-board timers via 82C54
- Two IEEE 959 SBX Bus interface connectors
- iLBX Bus support on i386 boards (no iLBX bus support on i486 CPU boards)
- Multiprocessor support, including memory aliasing
- Functional superset of iSBC 86/xx and iSBC 286/1x single board computers

i486™, i386™, i387™, Intel386™ and Intel486™ are trademarks of Intel Corporation.

October 1993
Order Number: 281110-003

intel

FEATURES

MICROPROCESSOR PERFORMANCE

Intel486™ DX2 and Intel486™ CPU Boards

The iSBC 486/12DX2S board features an Intel486™ DX2 microprocessor running at 66 MHz internal. The iSBC 486/12S CPU board features an Intel486 microprocessor running at 33 MHz. A 32-bit on-board system bus optimizes the compute performance of the i486 DX2 or i486 microprocessor. The same bus also provides support for up to 64 MB of zero wait-state parity DRAM.

Numeric processing applications are easily handled by the i486 on-chip floating point unit (FPU). The i486 FPU conforms fully to the ANSI/IEEE 754-1985 standard for floating point arithmetic and it provides full compatibility for applications written to the i387™ Numeric Coprocessor.

Intel386™ CPU Boards

The iSBC 386/12 CPU boards feature an i386 microprocessor running at 20 MHz. The on-board processor execution bus optimizes the compute performance of the i386 microprocessor. The on-board 32-bit memory bus provides support for up to 16 MB of zero wait-states (pipelined read hit) parity DRAM.

Numeric processing on the i386 CPU boards may be enhanced with the 32-bit i387 floating point math coprocessor. The i387 provides 80-bit precision, accelerating floating point calculations through hardware execution.

HIGH-SPEED DMA

Intel486™ DX2 and Intel486™ CPU Boards

2 MB to 64 MB of DRAM is supported via modules attached to the CPU baseboard. DRAM module access is optimized by a high-speed DMA controller and the latest fast-page DRAM technology. A DMA address generator (DAG) provides addressing for the full 4 gigabyte i486 memory address range. DRAM is dual-ported and addressable from the on-board 32-bit system bus or from the Multibus I bus.

Intel386™ CPU Boards

1 MB to 16 MB of DRAM is supported via modules attached to the CPU baseboard. DRAM module access is optimized by a high-speed DMA controller and the latest fast page DRAM technology. DRAM is dual-ported and fully addressable from the on-board 32-bit bus or from the Multibus I bus.

DUAL-BUS ARCHITECTURE

The iSBC 486/12DX2S, iSBC 486/12S and iSBC 386/12 CPU boards use multiple on-board busses, a design technique that increases overall system performance. The boards utilize both a high speed 32-bit execution bus for optimized CPU, memory, and math operations, and a dedicated I/O bus for on-board I/O, SBX bus I/O, and EPROM operations.

SCSI BUS INTERFACE

The iSBC 486/12DX2S, iSBC 486/12S, and iSBC 386/12S boards integrate a 10 MHz Fujitsu 87033B SCSI Protocol Controller (SPC). The 87033B SPC device supports a synchronous data rate of 5 MBytes/sec and an asynchronous rate of 2 MBytes/sec. SCSI data transfers are buffered into 16-bit wide. 512-bit deep bi-directional FIFOs and DMA transferred into (or from) local RAM at rates above the maximum incoming synchronous data rate. A 16-byte burst mode via a separate 16-byte burst FIFO is supported on the iSBC 486/12DX2S and the iSBC 486/12S. The SCSI implementation is ideal for applications requiring sustained high-speed SCSI data transfers.

COMPLETE ON-BOARD I/O FEATURE SET

The iSBC 486/12DX2S, iSBC 486/12S and iSBC 386/12 boards provide fully integrated on-board I/O functionality. Two serial ports are provided via Intel's 8247 Multi-Protocol Serial Controller. An 82C55A Programmable Peripheral Interface provides support for the parallel line printer interface. These boards also provide interrupt support based on two 82C59A Programmable Interrupt Controllers and the microprocessor's own non-maskable interrupt (NMI) line. Programmable timer/counter functions are provided by an 82C54 Programmable Interval Timer.

intel

FEATURES

Direct memory access transfers are provided by the 82258 advanced DMA (ADMA) controller. The DMA support is enhanced to increase flexibility and support synchronous transfers over the SCSI interface (iSBC 486/12DX2S, iSBC 486/12S, and iSBC 386/12S), serial ports (iSBC 386/12 and iSBC 386/12S only) and SBX Bus for high-speed I/O data transfers.

SBX BUS FOR I/O EXPANSION

Two IEEE 959 SBX Bus interface connectors are provided for I/O expansion. Functions such as additional serial/parallel ports, graphics, BITBUS, LAN interface, or analog to digital modules can be added onto the CPU boards via this industry standard I/O expansion interface. The SBX Bus interface is also ideal for integration of custom-built modules. The CPU boards support either two single-wide modules or one double-wide module.

MULTIBUS I, SBX, AND iLBX COMPLIANCE

The iSBC 486/12DX2S, iSBC 486/12S, and iSBC 386/12 boards comply with the published IEEE 796 Multibus specification (including full Multimaster) and the published IEEE 959 SBX specification.

The iSBC 386/12 and iSBC 386/12S boards support Multibus vectored interrupts and Intel's iLBX P2 bus. The iSBC 486/12DX2S, iSBC 486/12S, and iSBC 486/12 do not support these two capabilities.

Please see the specifications for more detail on Multibus, SBX, and iLBX bus compliance

MULTIPROCESSING SUPPORT

Multiprocessor support is enhanced through high-performance dual-port arbitration control logic and features such as memory aliasing over Multibus and real mode page registers. In addition, the addressing approach used for the iSBC 386/2X/3X series of boards is also supported.

SPECIFICATIONS

CPU

Intel486 DX2 Microprocessor @ 66 MHz

Intel486 Microprocessor @ 33 MHz

Intel386 Microprocessor @ 20 MHz

Intel387 Numeric Coprocessor @ 20 MHz

INTERRUPT CAPACITY

Interrupt sources: 26 total: On iSBC 486/12DX2 and iSBC 486/12 all 26 are jumper selectable; On iSBC 386/12, 5 are hard-wired to the 8259A PIC and 21 are jumper selectable.

Interrupt levels: 16 vectored requests using two 8259A devices and the CPU's NMI line.

Bus-vectored interrupts are supported only on the iSBC 386/12.

I/O CAPABILITY

Expansion: Two 8/16-bit SBX connectors, supporting up to two single-wide boards or one double-wide board.

Parallel: Line printer interface, on-board functions, and 3-bit board installed options code.

Serial: Two programmable channels using one 8274 device

SCSI: Single-ended, synchronous/asynchronous

Timers: Three programmable times using one 82C54 device

DMA: 10 MHz 82258 advanced DMA (ADMA) controller. Supports DMA block transfers between on-board memory over the Multibus I interface, SCSI interface (iSBC 486/12S and iSBC 386/12S), iLBX interface (iSBC 386/12 and iSBC 386/12S only), both SBX interfaces, and both serial channels (iSBC 386/12 and iSBC 386/12S only).

PHYSICAL CHARACTERISTICS

Width: 30.48 cm (12.00 in)
Height: 18.00 cm (7.05 in)
Depth: 2.18 cm (0.86 in), 4.11 cm (1.62 in) with added memory module

Recommended Slot spacing: 3.0 cm (1.2 in), 4.6 cm (1.8 in) for double-stacked memory modules.

intel

SPECIFICATIONS

ELECTRICAL CHARACTERISTICS

iSBC 486/12DX2 and iSBC 486/12DX2S DC Power Requirements (Maximum):

	Volts	Current 486/12DX2 486/12	Current 486/12DX2S 486/12S
Baseboard	+ 5.25	11.5A	13.5A
Baseboard	± 12	50 mA	50 mA
4 MByte DRAM Module	+ 5.25	2.8A	2.8A
8 MByte DRAM Module	+ 5.25	3.2A	3.2A

iSBC 386/12 and iSBC 386/12S DC Power Requirements (Maximum):

	Volts	Current
Baseboard	+ 5.25	14.5A
Baseboard	± 12	50 mA
1 MByte DRAM Module	+ 5.25	0.89A
2 MByte DRAM Module	+ 5.25	1.2A
4 MByte DRAM Module	+ 5.25	0.89A
8 MByte DRAM Module	+ 5.25	1.2A
i387™ Math Coprocessor	+ 5.25	0.31A

ENVIRONMENTAL CHARACTERISTICS

Operating Temperature: 0° to 55°C (32°F to 131°F) at 200 LFM airflow across board (default configuration)
Relative Humidity: 0% to 90% (without condensation)
Storage Temperature: −40°C to + 70°C (−40°F to +158°F)

ORDERING INFORMATION

REFERENCE MANUALS

507914 *iSBC 486/12S Single Board Computer Hardware Reference Manual.* Provides complete information on hardware features, installation, jumpering, memory maps, and addressing.

455528 *iSBC 386/12 Series Hardware Reference Manual.* Provides complete information on hardware features, installation, jumpering, memory maps, addressing, and schematics.

459913 *iSBC 386/12S Series Hardware Reference Manual.*

462435 *iSBC 386/12 Series Software Upgrade Guide.* Provides complete information on upgrading existing applications (hardware and software), from an iSBC 286/10A/12/14/16, iSBC 86/30/35, or iSBC 386/2X/3X to an iSBC 386/12 board.

i486™ SINGLE BOARD COMPUTERS

Part Number	Description
SBC48612F8A6	66 MHz with DMA, and 8 MB DRAM
SBC48612SF8A6	66 MHz with SCSI, DMA, and 8 MB DRAM
SBC48612F16A6	66 MHz with DMA, and 16 MB DRAM
SBC48612SF16A6	66 MHz with SCSI, DMA, and 16 MB DRAM
SBC48612SF32A6	66 MHz with SCSI, DMA, and 32 MB DRAM
SBC48612SF4A3	33 MHz with SCSI, DMA, and 4 MB DRAM
SBC48612SF8A3	33 MHz with SCSI, DMA, and 8 MB DRAM
SBC48612F8A3	33 MHz with DMA and 8 MB DRAM
SBC48612SF32A3	33 MHz with SCSI, DMA, and 32 MB DRAM
SBC48612SF16A3	33 MHz with SCSI, DMA, 16 MB DRAM

intel

ORDERING INFORMATION

i386™ SINGLE BOARD COMPUTERS

Pary Number	Description
SBC38612SF04	20 MHz with SCSI and 4 MB DRAM
pSBC38612SF08	20 MHz with SCSI and 8 MB DRAM
pSBC3861220F01	20 MHz with 1 MB DRAM
pSBC3861220F02	20 MHz with 2 MB DRAM
pSBC3861220F04	20 MHz with 4 MB DRAM
pSBC3861220F08	20 MHz with 8 MB DRAM

iSBC 386/12S models with an installed i387 coprocessor may be ordered by adding an M suffix to the above order codes. For example, pSBC3861220F08M is the order code for 20 MHz with 8 MB DRAM memory and the i387 math coprocessor installed.

MEMORY MODULES

i486™ Single Board Computers

Part Number	Description
SBCMM3008	8 MB DRAM module
SBCMM3016	16 MB DRAM module
SBCMM3032	32 MB DRAM module

i386™ Single Board Computers

Part Number	Description
pSBCMM01FP	1 MB DRAM module
pSBCMM02FP	2 MB DRAM module
pSBCMM04FP	4 MB DRAM module
pSBCMM08FP	8 MB DRAM module

For more information or the number of your nearest Intel Sales Office, call our Telesales Information Line at 800-438-4769 (good in the U.S. and Canada).

UNITED STATES
Intel Corporation
2200 Mission College Blvd.
Santa Clara, CA 95052-8119

JAPAN
Intel Japan K.K.
5-6 Tokodai, Tsukuba-shi
Ibaraki 300-26

FRANCE
Intel Corporation S.A.R.L.
1, Rue Edison, BP 303
78054 Saint-Quentin-en-Yvelines Cedex

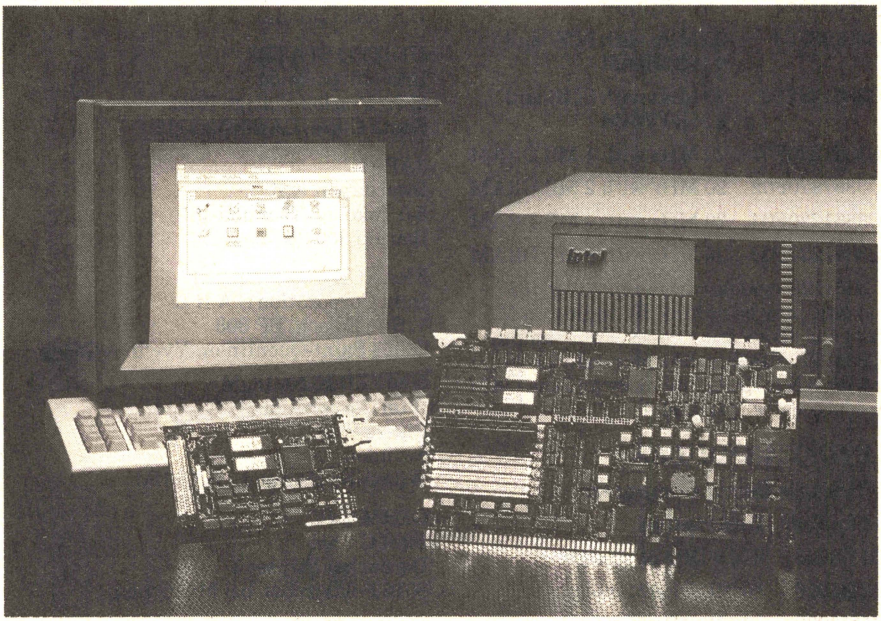
UNITED KINGDOM
Intel Corporation (U.K.) Ltd.
Pipers Way
Swindon
Wiltshire, England SN3 1RJ

GERMANY
Intel GmbH
Dornacher Strasse 1
8016 Feldkirchen bei Muenchen

HONG KONG
Intel Semiconductor Ltd.
10/F East Tower
Bond Center
Queensway, Central

CANADA
Intel Semiconductor of Canada, Ltd.
190 Attwell Drive, Suite 500
Rexdale, Ontario M9W 6H8

iSBC 386 SX* SINGLE BOARD COMPUTER AND iSBC 272* VGA GRAPHICS MODULE



281111-1

PC AT AND MULTIBUS I-COMPATIBLE i386™ SX CPU-BASED SINGLE BOARD COMPUTER*

The iSBC 386 SX single board computer is a cost-effective way to add 32-bit performance and DOS software compatibility to a Multibus I application. Based on the high-performance i386™ SX microprocessor and optional 387 SX Math Coprocessor, the iSBC 386 SX provides a 4X increase in compute power over most 8086 and 80186-based single board computers.

The iSBC 386 SX has the complete set of on-board I/O, real-time, and peripheral control resources found on a PC AT* motherboard and a Multibus I CPU board. Included are an on-board floppy controller, Winchester hard disk controller, two serial ports, parallel port, interrupt controllers, real-time clock, DMA and two full iSBX connectors. The optional iSBC 272 graphics module completes these capabilities by providing full VGA graphics.

*PC AT and EGA are registered trademarks of International Business Machines Corporation.
*The iSBC 386 SX and iSBC 272 are manufactured under product code pSBC 386 SX and pSBC 272 in Intel Puerto Rico II, Inc.
*Soft-Scope is a trademark of Concurrent Sciences, Inc.

August 1993
Order Number: 281111-001

FEATURES

FEATURES

- 16 MHz 386 SX CPU
- PC AT and DOS software compatibility
- Supports iRMX® for Windows and 32-bit real time operating system with DOS at the same time
- 387 SX Math Coprocessor socket
- 512K or 2 MB on-board DRAM, expandable to 8 MB
- Two 32-pin EPROM sockets plus two universal sockets for EPROM, SRAM, or FLASH
- On-board IDE hard disk and floppy controllers
- Two serial and one parallel ports
- Two 8/16-bit iSBX connectors
- Graphics add-on module supporting VGA, EGA*, CGA, Hercules Mono formats
- Complete on-board BIOS in EPROM

intel

FEATURES

HIGH-PERFORMANCE, COST-EFFECTIVE i386™ SX CPU PROVIDES EXCELLENT UPGRADE PATH

The iSBC 386 SX uses the powerful, cost-effective 386 SX microprocessor. The iSBC 386 SX supports an optional 387 SX Math Coprocessor for improved floating point calculations. To optimize compute performance, an on-board bus connects the 386 SX CPU, 387 SX Math Coprocessor, and local fast-page, 100ns, parity DRAM SIMM memory. A modular design helps reduce cost by providing sockets for the 387 SX and expandable DRAM memory so they may be selected to fit the needs of the design. The iSBC 386 SX is provided with 512K or 2 MB of local DRAM which may be expanded to 8 MB using plug-on SIMM packages.

iSBC 386 SX PROVIDES DOS PLUS REAL-TIME TO MICROCOMPUTER APPLICATIONS

The iSBC 386 SX is software compatible with the PC-AT motherboard, enabling users access to the large installed base of DOS software. The iSBC 386 SX comes ready to run, with an on-board BIOS that supports loading DOS from either the locally supported hard disk or floppy disks. The iSBC 386 SX can also run other i386 CPU-based software such as the iRMX III real-time operating system (O/S). This 32-bit operating system is ideal for controlling real-time processes common in industrial, communication, and other time-critical applications. The iRMX III software also enables DOS and the iRMX O/S to run concurrently on the board with full data exchange between the DOS and real-time applications. Using this software, engineers can design high performance real-time systems with easy-to-use DOS user interfaces.

HIGH INTEGRATION BOARD PROVIDES COMPLETE I/O FEATURE SET

First, the iSBC 386 SX includes the full set of I/O resources found on high-integration Multibus I single board computers, including two RS232 asynchronous serial ports, a Centronics-compatible parallel port, two integrated 8259 interrupt controllers, two 8/16 bit iSBX connectors, DMA and the Multibus interface. It also includes the full set of PC AT motherboard I/O resources, including an IDE interface to Winchester hard disks, interface to floppy drives, a real-time clock, and a keyboard and mouse interface. Traditional Multibus resources are shared with the AT features to provide interrupt and DMA services for the on-board I/O. Finally, the optional iSBC 272 VGA graphics module allows you to easily add VGA and extended VGA graphics to the iSBC 386 SX board.

ON-BOARD FEATURES HELP IN FITTING THE iSBC 386 SX INTO EXISTING MULTIBUS I DESIGNS

On-board features help you fit the iSBC 386 SX into existing Multibus I designs. The memory map includes access to the Multibus interface in memory space and I/O space. In memory space the board provides a 1 MB window, located at on-board address 13 MB and relocatable in your Multibus system memory space through use of a programmable register. Multibus I/O space is located at 8000H. If the board needs access to other intelligent boards through dual-ported memory, jumpers allow you to map SRAM or FLASH memory in the universal sites into the appropriate memory space required by your Multibus I system memory map (on one megabyte boundaries).

FEATURES

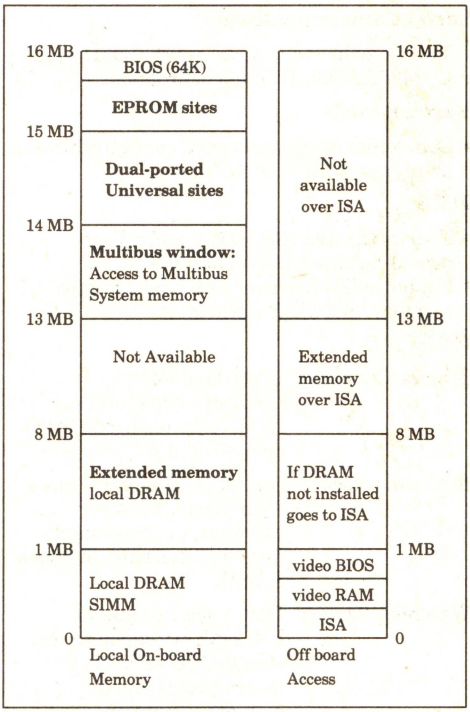


Figure 1: Multibus I Memory Map

ON-BOARD EPROM AND COMPLETE FEATURE SET INCREASE FLEXIBILITY

The iSBC 386 SX includes two 32-pin EPROM sockets for the BIOS and for addition of user-embedded code. Two additional 32-pin, universal JEDEC sites are provided for on-board, dual-ported memory. The sites support use of SRAM, FLASH or additional EPROM memory.

HIGH-INTEGRATION CUTS BOARD COUNT AND COSTS

The iSBC 386 SX board's complete set of I/O resources makes it possible to eliminate several boards from your design. For example, the iSBC 386 SX/iSBC 272 combination can take the place of the CPU board, peripheral controller board, graphics board, and any add-on modules for local memory or access to local printers, keyboard or console. Designs are simplified and cost-reduced, and performance is improved through a reduction in system bus traffic.

SOFTWARE AND DEVELOPMENT SUPPORT

To help complete system designs, Intel offers operating systems, compilers, debuggers, and other software development tools. Applications may be written in C, PL/M, FORTRAN, or assembler to run on DOS and/or the iRMX III operating system. Source-level debuggers include DB86 and Soft-Scope* III. In-circuit emulators and performance analysis tools are also available. To help develop your first system, Intel also offers a cable kit, converting the board's pin connectors to the appropriate industry-standard I/O connector.

ONE YEAR WARRANTY

All Intel board level products have a one year mail-in warranty on both parts and labor.

WORLD WIDE SERVICE AND SUPPORT

Should any Intel board ever need service, Intel maintains a world wide network of service and repair facilities to keep you and your customers up and running.

INTEL QUALITY — YOUR GUARANTEE

The iSBC 386 SX and iSBC 272 boards are designed and manufactured to meet Intel's strict standards, assuring their reliability and high quality.

SPECIFICATIONS

iSBC 386 SX SPECIFICATIONS

Computing and Memory Resources

- CPU: 386 SX microprocessor at 16 MHz
- Floating point math: 387 SX Numerics Coprocessor socket

DRAM Memory

- DRAM memory is fast page, 100 ns memory with parity
- DRAM Supplied:
  — iSBC386SX00 includes 512K using two 256 × 9 SIMM packages
  — iSBC386SX02 includes 2 MB using two 1Meg × 9 SIMM packages

Memory Performance

	Page "Hit"	Page "Miss"
READ	0 wait states	2 wait states
WRITE	1 wait state	1 wait state

Memory Expansion

- Sockets are provided for eight SIMM modules.
- Max DRAM available: 8 MB using 1Meg × 1 Meg SIMMs

EPROM and Other Local Memory

- Two 32-pin sites containing 128 Kbytes of EPROM memory with Phoenix BIOS. Sites will support 512 KB using 27020 EPROMs.
- Two additional 32-pin universal JEDEC sockets for additional EPROM, or for dual-ported SRAM or FLASH memory. Supports additional 512 KB using 27020 EPROMs.
- 114 bytes CMOS RAM for saving BIOS configurations, with provisions for off-board battery backup

I/O Resources

Hard Disk Controller

- Provides IDE interface to fixed disk drives.

Floppy Disk Controller

- Provides interface to the following floppy drives:
  1.44 MB or 720K 3.5″ floppy disk drive
  1.2 MB or 360K 5.25″ floppy disk drive

Serial Communications

- Two asynchronous serial ports, COM1 and COM2: RS232C DTE asynchronous

Parallel Port

- Centronics compatible port, configurable as a printer port LPT1 or LPT2.

DMA

- Two integrated 8237 DMA controllers providing seven DMA channels.
- Supports DMA between on-board floppy, memory, and iSBX connectors.

Other I/O

Timers/Counters:	integrated 8254 programmable interval timer/counter used for PC-compatibility
Real-time clock:	complete time-of-day clock with alarm, 100-year calendar, programmable periodic interrupt, 50 bytes SRAM.
Watchdog Timer:	Dallas semiconductor DS1232S watchdog timer. Resets CPU on timeout (every 600ms) once activated. Jumper-selectable.
Battery back-up connectors:	connectors are provided to off-board battery back-up via the P2 connector or via an on-board connector.

Interrupt Capacity

- Two integrated 8259 PICs
- 15 interrupts, 8 for Multibus interface

SPECIFICATIONS

Interfaces/Connectors

Hard disk:	40-pin header
Floppy Disk:	connector (J4), 34-pin header
Serial port COM1:	10-pin, two row header
Serial port COM2:	10-pin, two row header
Parallel port:	26-pin, two row header
Keyboard/Mouse share same connector	10-pin, two row header
Battery back-up:	using either P2, or using four on-board stake pins
Speaker:	two stake pins for connecting a piezo speaker
Multibus I (IEEE796) Bus Compliance:	Master D16 M24 I16 VOL SLAVE D16 M24
iSBX Bus Compliance:	D16/16 DMA I

iSBC 272 SPECIFICATIONS

Graphics Protocols Supported

- Software protocols: VGA, extended VGA, EGA, CGA, Hercules monochrome

Resolution

- Standard VGA modes: 640×480, 16 colors, graphics; 720×400, text
- Extended VGA modes: 800×600, 16 colors, graphics; 1024×768, 4 colors, graphics

Memory

- 256K DRAM

Connectors

- Ten-pin header to analog cable for VGA monitors

iSBC 386 SX AND iSBC 272 SPECIFICATIONS

Software Support

- Operating systems: DOS, MS-DOS, iRMXIII, and DOS/iRMX operating systems

PHYSICAL CHARACTERISTICS

Physical Dimensions

Width and length conform to Multibus I specifications.
Depth: 1.00", 1.62" with iSBC272 module.

ELECTRICAL CHARACTERISTICS

DC Power Requirements (Typical):

	Current
iSBC 386 SX	6A
iSBC 272	1A

ENVIRONMENTAL CHARACTERISTICS

Operating Temperature: 0°C to 55°C at 200 LFM airflow

Relative Humidity: 0% to 90% (without condensation)

Storage Temperature: −40°C to +70°C

REFERENCE MANUALS

iSBC 386 SX Hardware User's Manual, Order Number: 507913-001

intel

ORDERING INFORMATION

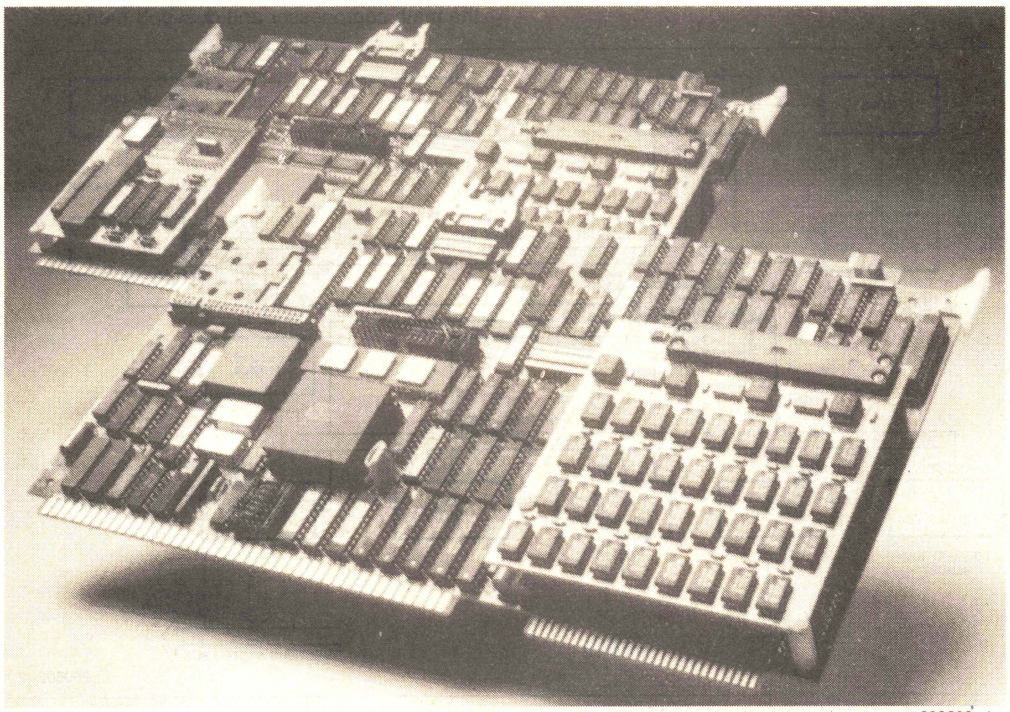
Part Number	Description
SBC386SX00	386 SX microprocessor-based, PC AT compatible single board computer with 512 Kbytes on-board DRAM.
SBC386SX02	Same as iSBC386SX00 with 2 MB on-board DRAM.
SBC272	VGA graphics module for use with iSBC386SX boards.
CABLESBC386SX	Complete set of cables for hard disk, floppy, two serial ports, parallel port, keyboard/mouse, and VGA analog interfaces. Cables convert pin-header connectors to industry-standard, PC-compatible connectors.

For more information or the number of your nearest Intel sales office, call 800-548-4725 (good in the U.S. and Canada.)

iSBC 386/21/22/24/28 AND 386/32/34/38* SINGLE BOARD COMPUTERS

- **Choice of 16 MHz or 20 MHz i386™ Microprocessor**
- **Available with 1, 2, 4, or 8 Megabytes of On-Board 32-Bit Memory, expandable to 16 Megabytes**
- **High Speed 80387 Floating Point Math Coprocessor**
- **Two 32-Bit JEDEC Sites for up to 512 Kilobytes of EPROM Memory**
- **RS232C Interface for Local/Remote Control and Diagnostics**
- **iSBX Interface for Low Cost I/O Expansion**
- **16 Levels of Direct Vectored Interrupt Control**
- **64 Kilobyte 0 Wait-State Cache Memory**

The iSBC 386/2x and 3x series boards (iSBC 386/21/22/24/28 and iSBC 386/32/34/38) feature either a 16 MHz or 20 MHz 386 CPU, an 80387 math coprocessor, a 64k byte, 0 wait-state cache memory to support the CPU, and a 32-bit interface to 1, 2, 4, or 8 megabytes of dual-port parity DRAM memory. An additional 1, 2, 4, or 8 MB iSBC MM0x series memory module may be installed to provide up to 16 MB of on-board DRAM memory. The iSBC 386/2x and 3x boards also feature an 8/16-bit iSBX MULTIMODULE interface for low-cost I/O expansion, an asynchronous RS232C interface to support a local terminal or modem, two 16-bit programmable timer/counters, a 16-level direct-vectored interrupt controller, two 32-pin JEDEC sites and multimaster Multibus arbitration logic.



280602-1

*The iSBC 386/21/22/24/28 and iSBC 386/32/34/38 Boards are manufactured under product code pSBC 386/21/22/24/28 and pSBC 386/32/34/38 by Intel Puerto Rico, Inc.

*XENIX is a registered trademark of Microsoft Corp.
*UNIX is a trademark of AT&T.

October 1993
Order Number: 280602-003

OVERVIEW—iSBC 386/2x AND 3x SERIES CPU BOARDS

The iSBC 386/21/22/24/28 and iSBC 386/32/34/38 boards (iSBC 386/2x and 3x series) are 16 MHz and 20 MHz versions of Intel's first Multibus I i386 microprocessor CPU boards. The boards employ a dual-bus structure, a 32-bit CPU bus for data transfers between the CPU and memory, and a 16-bit bus for data transfers over the MULTIBUS interface, iSBX interface, EPROM local memory, and I/O interfaces. The boards take advantage of the i386 CPU's 32-bit performance while maintaining compatibility with the Multibus I interface.

The DRAM memory, which is on a module that is secured to the baseboard, may be expanded by installing a second 1, 2, 4, or 8M byte memory module.

16 MHz or 20 MHz Central Processor Unit

The heart of the iSBC 386/2x and 3x CPU board is the i386 microprocessor. The complete series includes two lines, with a choice of CPU speed. The iSBC 386/21/22/24/28 boards use the 16 MHz i386 microprocessor and the iSBC 386/32/34/38 boards use the 20 MHz i386 microprocessor. The i386 CPU utilizes address pipelining, a high speed execution unit, and on-chip memory management/protection to provide the highest level of system performance. The i386 microprocessor also features an Address Translation Unit that supports up to 64 terabytes of virtual memory.

The i386 CPU is upward compatible from Intel's 8088, 8086, 80186, and 80286 CPUs. Application software written for these other 8- and 16-bit microprocessor families can be recompiled to run on the 80386 microprocessor. Some changes to the software such as adjustment of software timing loops and changing I/O address references may be required. The 386 microprocessor resides on the 32-bit wide CPU bus which interconnects the CPU with the math coprocessor and dual-port memory.

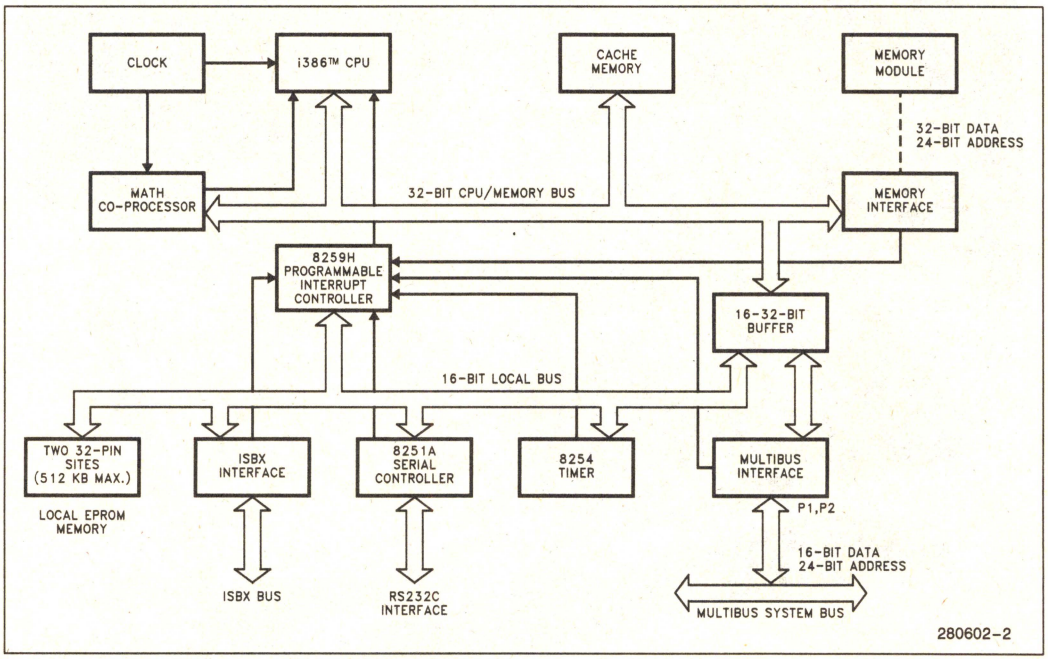


Figure 1. iSBC 386/2x and 3x CPU Board Block Diagram

Instruction Set

The i386 CPU instruction set includes: variable length instruction format (including double operand instructions; 8-, 16-, and 32-bit signed and unsigned arithmetic operators for binary, BCD and unpacked ASCII data; and iterative word and byte string manipulation functions. All existing instructions have been extended to support 32-bit addresses and operands. New bit manipulation and other instructions have been added for extra flexibility in designing complex software.

Numeric Data Processor

For enhanced numerics processing compatibility, the iSBC 386/2x and 3x boards include an 80387 numeric coprocessor. Over 60 numeric instructions offer arithmetic, trigonometric, transcendental, logarithmic and exponential instructions. Supported data types include 16-, 32-, and 64-bit integer, 32- and 64-bit floating point, 18-digit packed BCD and 80-bit temporary. Data transfers to and from the CPU are 32-bits wide.

Architectural Features

The 8086, 8088, 80188, 80286, and i386 microprocessor family contain the same basic sets of registers, instructions, and addressing modes. The i386 processor is upward compatible with the 8086, 8088, 80186, 80188, and 80286 CPU's.

Architectural Features

The i386 CPU operates in two modes: protected virtual address mode; and 8086 real address mode. In protected virtual address mode (also called protected mode), programs use virtual addresses. In this mode, the i386 CPU automatically translates logical addresses to physical addresses. This mode also provides memory protection to isolate the operating system and ensure privacy of each task's programs and data. In 8086 real address mode, programs use real addresses with up to one megabyte of address space. Both modes provide the same base instruction set and registers.

Interrupt Control

Incoming interrupts are handled by two cascaded on-board 8259A programmable interrupt controllers and by the i386 CPU's NMI line. Twenty interrupt sources are routed to the programmable controllers and the interrupt jumper matrix. Using this jumper matrix, the user can connect the desired interrupt sources to specific interrupt levels. The interrupt controllers prioritize interrupts originating from up to 15 sources and send them to the CPU. The user can connect a sixteenth interrupt to the i386 NMI line. Table 1 includes a list of devices and functions suported by interrupts. Bus vectored interrupts are not supported.

Table 1. Interrupt Request Sources

Source	Function	Number of Interrupts
Multibus Interface	Requests from Multibus resident peripherals or other CPU boards	8
8251A Serial Controller	Indicates status of transmit and receive buffers and RI lead of the RS232C interface	3
8254 Timers	Timer 0, 1 outputs; function determined by timer mode (hardwired to interrupt controller)	2
iSBX Connector	Function determined by iSBX MULTIMODULE board	4
Bus Timeout	Indicates addressed Multibus or iSBX resident device has not responded to a command within 10 ms	1
Power Fail Interrupt	Indicates AC power is not within tolerance (signal generated by system power supply)	1
Parity Interrupt	Indicates on-board parity error	1
Programmable Register	Generate interrupt under program control	1

Memory Capabilities

The iSBC 386/2x and 3x boards support both EPROM local memory and dynamic RAM (DRAM), which is located on-board. The DRAM is supported by a high speed on-board cache memory.

DRAM Memory

The iSBC 386/2x and 3x series CPU boards come with 1, 2, 4, or 8M bytes of DRAM memory. This memory is on a low profile module that is installed on the baseboard. The module measures approximately 4″ x 4″ and uses surface mount DRAM devices. The DRAM memory supports byte-parity error detection and has a 32-bit wide data path to the 80386 CPU and 16-bit wide data path to the Multibus interface.

The memory may be expanded by installing an additional iSBC MM0x series memory module, which is available in 1, 2, 4, or 8M byte sizes. All mounting hardware is included. Maximum DRAM memory is 16M bytes using an iSBC 386/28 or 386/38 CPU board and an 8M byte iSBC MM08 memory module. This combination requires 1.8 inches of cardcage space.

Cache Memory

A 64K byte cache memory on the iSBC 386/2x and 3x boards supports the i386 CPU and provides 0 wait-state reads for data and program code resident in the cache memory. The cache memory is updated whenever data is written into the dual-port memory or when the CPU executes a read cycle and the data or program code is not present in cache memory. This process is controlled by the cache replacement algorithm. Cache "misses" require additional wait-states to retrieve data from the DRAM memory. If the processor is in pipelined mode, 2 wait-states (4 clock cycles) are required to retrieve data. If the processor is in non-pipelined mode, 3 wait-states are required. All writes to DRAM memory require 2 (pipelined) or 3 (non-pipelined) wait-states.

The cache memory supports 16K entries, with each entry comprised of a 32-bit data field and an 8-bit tag field. The tag field is used to determine which actual memory word currently resides in a cache entry. The cache memory size and effective replacement algorithm are designed to optimize both the probability of cache "hits" and local bus utilization.

EPROM Memory

The EPROM memory consists of two 32-pin JEDEC sites that are intended for boot-up and system diagnostic/monitor routines, application code, and ROM-able operating system software. Maximum local memory capacity is 512K bytes using Intel 27020 (256k x 8) 2 megabit EPROM devices. The EPROM memory resides at the upper end of the 386 device's memory space for both real address mode and PVAM operation.

Memory Map

In real address mode, the maximum amount of addressable physical memory is 1 Mbyte. In protected virtual address mode (PVAM), the maximum amount of addressable physical memory is 16 Mbytes. The system designer can easily change the CPU memory map to adapt the CPU board to the required overall system memory map. Reconfiguration is usually necessary for multiple processor-based systems with more than two CPU boards and/or intelligent I/O boards. By changing PAL devices and/or by moving jumpers, the designer can set:

- EPROM memory space
- Starting address of DRAM memory
- Amount of DRAM memory that is dual-ported to the CPU and Multibus interface or single-ported to the CPU
- Access to off-board Multibus address space

EPROM Memory

The EPROM memory space is set using four jumpers to accommodate 27256 (256 kb), 27512 (512 kb), 27010 (1 Mb), or 27020 (2 Mb) byte-wide devices. Smaller EPROM devices may be used, however the EPROM will appear more than once within the EPROM address space. Using a pair of 27020 EPROMs will provide 512k bytes of memory. The iSBC 386/2x and 3x series boards are designed to accommodate EPROM devices with access times ranging from 130 ns–320 ns. In real address mode, the ending address of EPROM memory is always 1M byte (FFFFFH). In PVAM, the ending address of EPROM memory is always 4G bytes (FFFF FFFFH), which is the top of the 386 CPU address space.

DRAM Memory Size/Location

The iSBC 386/2x and 3x boards allow the user to control the location and size of the DRAM memory (on the iSBC 386/2x and 3x boards) available for use by the CPU and other boards in the system. In PVAM, the starting address of DRAM can be set to start on any 1M byte boundary up through 15M bytes by setting jumpers and by installing a custom-programmed PAL device. In real address mode, the DRAM memory always starts at 0H (hex).

The ending address can be set on 64k byte boundaries using jumpers in both PVAM and real address mode. Setting the ending address at lower than the actual amount of installed memory effectively deselects a portion of DRAM and creates additional Multibus address space.

Multibus Address Space

Any address space not set aside as EPROM or DRAM memory automatically becomes address space the CPU can use to access other boards in the system. For example, Figure 2A shows a real address mode CPU memory map for a 1M byte iSBC 386/21 board. With the DRAM ending address set at 512k bytes and 128k bytes of installed EPROM, 384k bytes of Multibus address space is accessable by the CPU. Figure 2B shows a typical PVAM configuration where the 4 Mbytes of DRAM has been set to start at 1M byte and end at 4.5M bytes. The address space from 0 to 1M byte and 4.5 to 16M bytes is the Multibus address space accessable by the CPU.

Figure 2C illustrates another way the board can establish additional Multibus address space. If the DRAM memory starts at 0, a jumper on the board can be used to create additional Multibus address space between 512k bytes and 1M byte. This feature is available both in real address mode and PVAM.

Dual-Port/Local Memory

A portion or all of the DRAM memory can be selected to be dual-port (shared) memory. Both the starting and ending addresses are set on 256k byte boundaries using jumpers on the board. Any DRAM memory that is not configured as dual-port memory is local (single-port) memory available only to the CPU.

Programmable Timer

Three 16-bit, programmable interval timer/counters are provided using an 8254 device, with one timer dedicated to the serial port for use as a baud rate generator. The other two timers can be used to generate accurate time intervals under software control. The timers are not cascadable. Four timer/counter modes are available as listed in Table 2. Each counter is capable of operating in either BCD or binary modes. The contents of each counter may be read at any time during system operation.

Table 2. Programmable Timer Functions

Function	Operation
Interrupt on terminal count	When terminal count is reached, an interrupt request is generated. This function is extremely useful for generation of real-time clocks.
Rate generator	Divide by N counter. The output will go low for one input clock cycle, and the period from one low going pulse to the next is N times the input clock period.
Square-wave rate generator	Output will remain high until one-half the count has been completed, and go low for the other half of the count.
Software triggered strobe	Output remains high until software loads count (N). N counts after count is loaded, output goes low for one input clock period.

Serial I/O

The iSBC 386/2x and 3x boards include one RS232C serial channel, which is configured as an

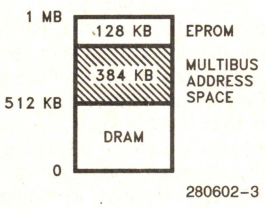


Figure 2A. Real Address Mode iSBC 386/21 Board Memory Map

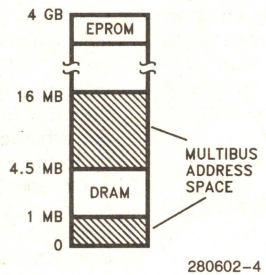


Figure 2B. PVAM iSBC 386/24 Board Memory Map

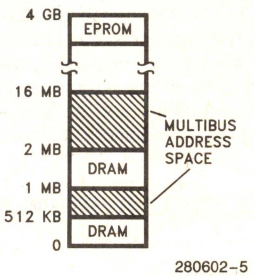


Figure 2C. PVAM iSBC 386/22 Board Memory Map

asynchronous, DTE interface. Data rates up to 19.2k baud may be selected. The serial channel can connect either to a host system for software development or to a stand alone terminal for field diagnostic support. For stand alone use, unhosted monitor software needs to be programmed by the user into the local EPROM memory. The serial channel may also be connected to a modem to provide remote diagnostic support or to download program codes. The physical interface is a 10-pin ribbon-style connector located on the front edge of the board.

iSBX Interface

For iSBX MULTIMODULE support, the iSBC 386/2x and 3x CPU boards provide an 8/16-bit iSBX connector that may be configured for use with either 8- or 16-bit, single or double-wide iSBX MULTIMODULE boards. Using the iSBX interface, a wide variety of specialized I/O functions can be added easily and inexpensively to the iSBC 386/2x and 3x boards.

Reset Functions

The iSBC 386/2x and 3x boards are designed to accept an Auxilliary Reset signal via the boards' P2 interface. In this way, system designs that require front panel reset switches are supported. The iSBC 386/2x and 3x boards use the AUX reset signal to reset all on-board logic (excluding DRAM refresh circuitry) and other boards in the Multibus system. The iSBC 386/2x and 3x boards will also respond to an INIT reset signal generated by another board in the system.

LED Status Indicators

Mounted on the front edge of the iSBC 386/2x and 3x boards are four LED indicators that indicate the operating status of the board and system. One LED is used to show that an on-board parity error or a Multibus bus parity error has occurred. A second LED indicates that a Multibus or iSBX bus access timeout has occurred. The third LED is triggered by the start of an 386 bus cycle and will turn off if the 386 CPU stops executing bus cycles. The fourth LED will light under software control if the program writes to a specific I/O location.

MULTIBUS SYSTEM ARCHITECTURE

Overview

The Multibus system architecture includes three bus structures: the Multibus system bus, the iLBX local bus extension and the iSBX MULTIMODULE expansion bus. Each bus structure is optimized to satisfy particular system requirements. The system bus provides a basis for general system design including memory and I/O expansion as well as multiprocessing support. The iLBX bus, which is usually used for memory expansion, is not supported by the iSBC 386/2x and 3x boards since all DRAM memory is located on-board. The iSBX bus povides a low cost way to add I/O to the board.

System Bus—IEEE 796

The Multibus system bus is an industry standard, IEEE 796, microcomputer bus structure. Both 8- and 16-bit single board computers are supported on the IEEE 796 structure with 24 address and 16 data lines. In its simplest application, the system bus allows expansion of functions already contained on a single board computer (e.g., memory and digital I/O). However, the IEEE 796 bus also allows very powerful distributed processing configurations using multiple processors, I/O boards, and peripheral boards. The Multibus system bus is supported with a broad array of board level products, VLSI interface components, detailed published specifications and application notes.

System Bus—Expansion Capabilities

The user can easily expand or add features to his system by adding various Multibus boards to his system. Products available from Intel and others include: video controllers; D/A and A/D converter boards; peripheral controller cards; communications/networking boards; voice synthesis and recognition boards; and EPROM memory expansion boards.

System Bus—Multimaster Capabilities

For those applications requiring additional processing capacity and the benefits of multiprocessing (i.e., several CPUs and/or controllers sharing system tasks through communication over the system bus), the iSBC 386/2x and 3x boards provide full system bus arbitration control logic. This control logic allows up to four bus masters to share the system bus using a serial (daisy chain) priority scheme. By using an external parallel priority decoder, this may be extended to 16 bus masters. In addition to multiprocessing, the multimaster capability also provides a very efficient mechanism for all forms of DMA (Direct Memory Access) transfers.

iSBX Bus MULTIMODULE On-Board Expansion

Through this interface, additional on-board I/O functions may be added, such as parallel and serial I/O, analog I/O, small mass storage device controllers (e.g., floppy disks), BITBUS Control, and other custom interfaces to meet specific needs. Compared to other alternatives such as Multibus I boards, iSBX modules need less interface logic and power, and offer simpler packaging and lower cost. The iSBX interface connector on the iSBC 386/2x and 3x boards provides all the signals necessary to interface to the local on-board bus, and is compatible with both 8-bit and 16-bit MULTIMODULES.

SOFTWARE SUPPORT

Operating Systems

The iSBC 386/2x and 3x boards are supported by a variety of operating systems, including the iRMX® I, iRMX II, iRMX III, XENIX and System V/386 operating systems (third party vendors).

See the datasheets on these software products for further details.

BOARD SPECIFICATIONS

Word Size

Instruction—8, 16, 24, 32 or 40 bits
Data—8, 16, 32 bits

System Clock

386 CPU—16 MHz or 20 MHz
Numeric Processor—80387 module—16 MHz or 20 MHz

DRAM Memory

On-board parity memory
- iSBC 386/21 board—1M byte
- iSBC 386/22/32 board—2M bytes
- iSBC 386/24/34 board—4M bytes
- iSBC 386/28/38 board—8M bytes

Memory expansion—One additonal plug-in module:
- iSBC MM01FP—1M byte
- iSBC MM02FP—2M bytes
- iSBC MM04FP—4M bytes
- iSBC MM08FP—8M bytes

Maximum Addressable Physical Memory—16 Megabytes (protected virtual address mode) 1 Megabyte (real address mode)

EPROM Memory

Number of sockets—Two 32-pin JEDEC Sites (compatible with 28-pin and 32-pin devices)

Sizes accommodated—64 kb (8k x 8), 128 kb (16k x 8), 256 kb (32k x 8), 512 kb (64k x 8), 1 Mb (128k x 8), 2 Mb (256k x 8)

Device access speeds—130 ns to 320 ns
Maximum memory—512k bytes with 27020 (2M bit) EPROMs

I/O Capability

Serial Channel

Type—One RS232C DTE asynchronous channel using an 8251A device

Data Characteristics—5-8 bit characters; break character generation; 1, 1½, or 2 stop bits; false start bit detection; automatic break detect and handling; even/odd parity error generation and detection

Speed—110, 150, 300, 600, 1.2 kb, 2.4 kb, 4.8 kb, 9.6 kb, 19.2 kb

Leads supported—TD, RD, RTS, CTS DSR, RI, CD, SG

Connector Type—10 pin ribbon

Expansion—One 8/16-bit iSBX interface connector for single or double wide iSBX MULTIMODULE board.

Interrupt Capacity

Potential Interrupt Sources—21 (2 fixed, 19 jumper selectable)

Interrupt Levels—16 using two 8259A devices and the 80386 NMI line

Timers

Two programmable timers using one 8274 device.

Output Frequencies/Timing Intervals

Function	Single Counter	
	Min	Max
Real-time interrupt	1.63 μs	53.3 ms
Rate Generator	18.8 Hz	615 kHz
Square-wave rate generator	18.8 Hz	615 kHz
Software triggered strobe	1.63 μs	53.3 ms

Interfaces

Multibus Bus—All signals TTL compatible
iSBX Bus—All signals TTL compatible
Serial I/O—RS 232C, DTE

MULTIBUS DRIVERS

Function	Type	Sink Current (mA)
Data	Tri-State	64
Address	Tri-State	24
Commands	Tri-State	32
Bus Control	Open Collector	16/32

Power Requirements

iSBC 386/2x and 3x boards
Maximum: +5V, 12.5A
±12V, 35 mA
Typical: +5V, 9A
±12V, 20 mA

NOTE:
Does not include power for iSBX module, EPROM memory, or added iSBCMM0x memory modules.

Add the following power when adding iSBC MM0X memory modules:

iSBC MM01FP +5V, 0.71A
MM02FP +5V, 0.96A
MM04FP +5V, 0.71A
MM08FP +5V, 0.96A

Environmental Requirements

Operating Temperature—0°C to 60°C at 300 LFM
Relative Humidity—0% to 85% noncondensing
Storage Temperature—−40°C to +70°C

Physical Characteristics

Dimensions
Width—12.00 in. (30.48 cm)
Height—7.05 in. (17.91 cm)
Depth—0.86 in. (2.18 cm), 1.62 in. (4.11 cm) with added memory module

Recommended Minimum Cardcage Slot Spacing
1.2 in. (3.0 cm), with or without iSBX MULTIMODULE
1.8 in. (4.6 cm), with addded iSBC MM0x memory module

Approximate Weight
26 oz. (738 gm)

Reference Manual

149094—iSBC 386/21/22/24/28 Hardware Reference Manual (order separately)

453652—iSBC 386/32/34/38 Single Board Computer Hardware Reference Manual

Ordering Information

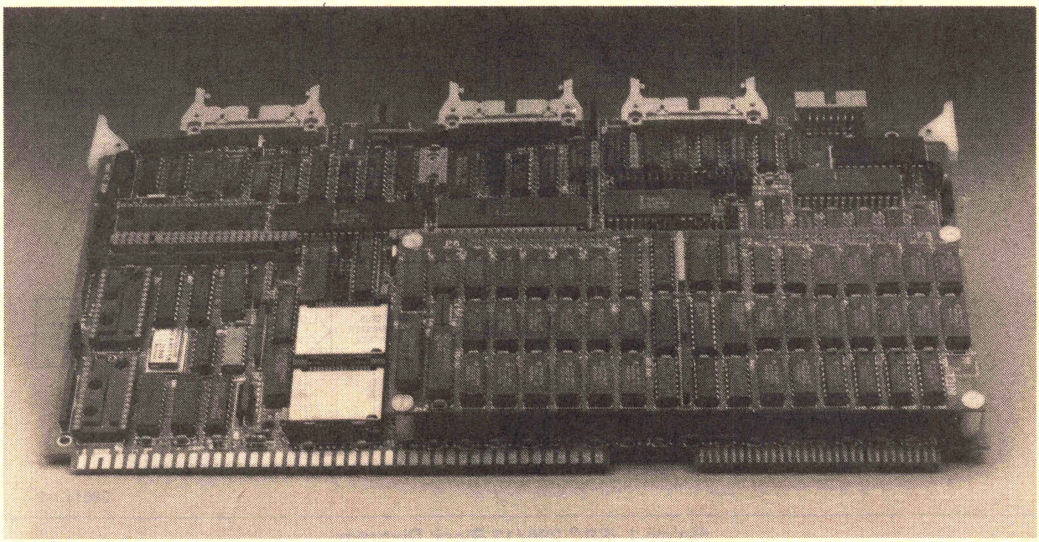
Part Number	Description
CPU Boards	
SBC38621	16 MHz 386 Multibus I CPU Board with 1 MB DRAM Memory
SBC38622	16 MHz 386 Multibus I CPU Board with 2 MB DRAM Memory
SBC38624	16 MHz 386 Multibus I CPU Board with 4 MB DRAM Memory
SBC38628	16 MHz 386 Multibus I CPU Board with 8 MB DRAM Memory
SBC38632	20 MHz 386 Multibus I CPU Board with 2 MB DRAM Memory
SBC38634	20 MHz 386 Multibus I CPU Board with 4 MB DRAM Memory
SBC38638	20 MHz 386 Multibus I CPU Board with 8 MB DRAM Memory
Memory Modules	
SBCMM01FP	1 MB Parity DRAM Memory Expansion Module
SBCMM02FP	2 MB Parity DRAM Memory Expansion Module
SBCMM04FP	4 MB Parity DRAM Memory Expansion Module
SBCMM08FP	8 MB Parity DRAM Memory Expansion Module

iSBC 286/12, 286/14, 286/16* SINGLE BOARD COMPUTERS

- 8 MHz 80286 Microprocessor
- Two JEDEC 28-Pin Sites for up to 128K Bytes of Local EPROM Memory, Expandable to 256K Bytes Using an iSBC 341 Expansion Module
- 1, 2, or 4 Megabyte, 0 Wait-State, Dual-Port, Parity Memory
- Supports User Installed 80287 Numeric Data Processor and 82258 Advanced DMA Controller Devices
- Two iSBX Bus Interface Connectors for I/O Expansion
- Synchronous High-Speed Interface for 0 Wait-State Read/Write to EX Memory Expansion Boards
- iLBX Interface for iLBX Memory Board Expansion
- 16 Levels of Vectored Interrupt Control
- Centronics-Compatible Parallel I/O Printer Interface
- Two Programmable Multiprotocol Synchronous/Asynchronous Serial Interfaces; One RS232C, the Other RS232C or RS422/449 Compatible

The iSBC 286/12, iSBC 286/14, and iSBC 286/16 Single Board Computers are members of Intel's family of 16-bit microcomputers. The boards feature an 80286 microprocessor running at 8 MHz together with 1, 2, or 4 megabytes of dual-ported, 0 wait-state, parity memory. These features make the iSBC 286/12/14/16 boards the ideal single board solution for applications requiring high performance and up to 1, 2, or 4 megabytes of memory. For those applications needing more memory, up to four memory expansion boards may be connected to the iSBC 286/12/14/16 boards over its P2 interface. The P2 interface supports both standard iLBX memory boards and Intel's EX series of synchronous, 0 wait-state, memory boards that provide up to 16 megabytes of system memory. The iSBC 286/12/14/16 boards also feature two sockets for user installed 80287 Numeric Data Processor and 82258 Advanced Direct Memory Access Controller devices. These components further increase board performance by off-loading time intensive tasks from the 80286 microprocessor. The iSBC 286/12/14/16 CPU boards are true single-board solutions that also include two serial I/O channels, one parallel line printer channel, local memory, interrupt controllers and programmable timers all on one board.

*The iSBC 286/12; iSBC 286/14 and iSBC 286/16 are also manufactured under product code pSBC28612; pSBC28614 and pSBC28616 by Intel Puerto Rico, Inc.



280147-1

*XENIX is a registered trademark of Microsoft Corp.
**UNIX is a trademark of UNIX System Laboratories.

August 1993
Order Number: 280147-004

FUNCTIONAL DESCRIPTION

Overview

The iSBC 286/12/14/16 boards utilizes the 80286 CPU within the Multibus system architecture, enhanced by the industry standard iLBX bus and a new, 0 wait-state, synchronous memory interface, to provide a high-performance 16-bit solution. This board features 1, 2, or 4 megabytes of dual-port, 0 wait-state, parity memory, plus interrupt, memory and I/O features facilitating a complete single-board computer system. The iSBC 286/12/14/16 boards can be used in many applications originally designed for Intel's other 16-bit microcomputers.

Central Processing Unit

The central processor for the iSBC 286/12/14/16 board is the 80286 CPU operating at an 8.0 MHz clock rate. The 80286 CPU is upwardly compatible with Intel's 8088 and 8086 CPUs. The 80286 CPU runs 8088 and 8086 code at substantially higher speeds due to its parallel architecture. In addition, the 80286 CPU provides on-chip memory management and protection and virtual memory addressing of up to 1 gigabyte per task. Processing speed and efficiency may be further enhanced by installing an 80287 numerics coprocessor and an 82258 ADMA controller. The clock rates for the 80286 and the 80287 are independent with the 80287 rate jumper selectable at either 5.3 MHz or 8.0 MHz.

Instruction Set

The 80286 instruction repertoire includes variable length instruction format (including double operand instructions), 8-bit and 16-bit signed and unsigned arithmetic operators for binary, BCD and unpacked ASCII data, and iterative word and byte string manipulation functions.

Numeric Data Processor

For enhanced numerics processing capability, the 80287 Numeric Data Processor extends the 80286 architecture and data set. Over 60 numeric instructions offer arithmetic, trigonometric, transcendental, logarithmic and exponential instructions. Supported data types include 16-, 32-, and 64-bit integer, 32- and 64-bit floating point, 18-digit packed BCD and 80-bit temporary.

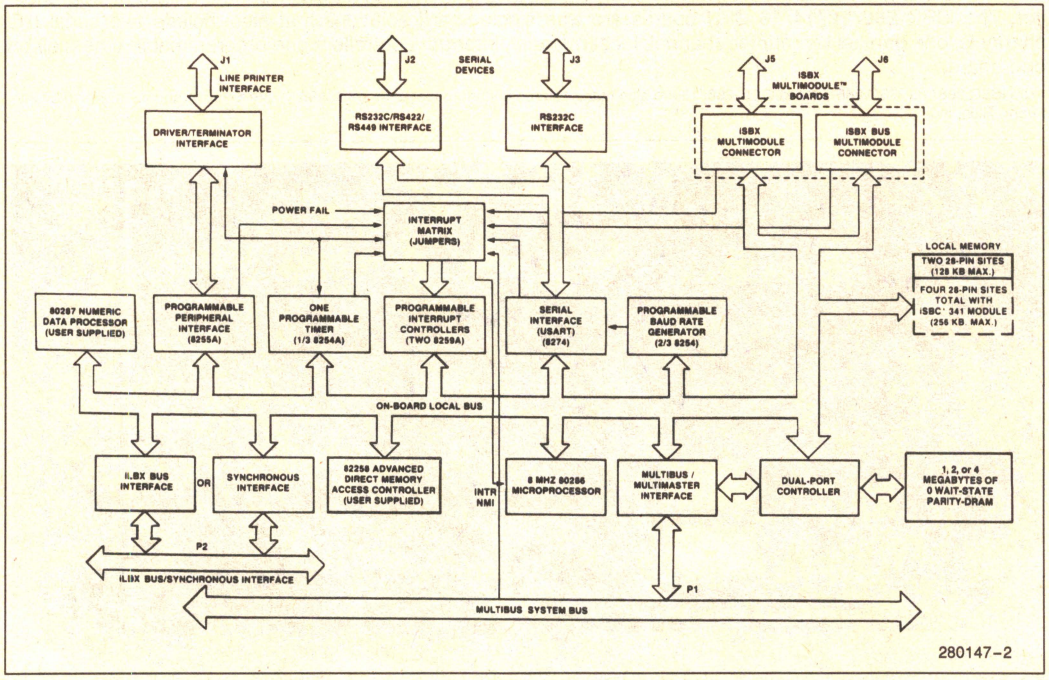


Figure 1. iSBC 286/12 Block Diagram

Advanced DMA Controller

For those applications that require frequent moving of large blocks of data, the user may install an Intel 82258, 4 channel, advanced DMA (ADMA) controller to further increase system performance. The ADMA Controller supports DMA requests from the 8274 USART (2 channels) and the iSBX interfaces on the board (1 per interface). The ADMA can also perform data transfers over the on-board CPU bus, the MULTIBUS (P1) interface, and the iLBX/synchronous (P2) interface.

ARCHITECTURAL FEATURES

The 8086, 8088, 80186 and 80286 microprocessor family contains the same basic set of registers, instructions, and addressing modes. The 80286 processor is upward compatible with the 8086, 8088, and 80186 CPUs.

The 80286 operates in two modes: 8086 real address mode, and protected virtual address mode. In 8086 real address mode, programs use real address with up to one megabyte of address space. Programs use virtual addresses in protected virtual address mode, also called protected mode. In protected mode, the 80286 CPU automatically maps 1 gigabyte of virtual addresses per task into a 16 megabyte real address space. This mode also provides memory protection to isolate the operating system and ensure privacy of each task's programs and data. Both modes provide the same base instruction set and registers.

Vectored Interrupt Control

Incoming interrupts are handled by two on-board 8259A programmable interrupt controllers (PIC) and by the 80286's NMI line. Interrupts originating from up to 15 sources are prioritized and then sent to the CPU. The 8259A devices support both polled and vectored mode of operation. Further interrupt capability is available through bus vectored interrupts where slave 8259A interrupt controllers resident on separate iSBC Boards supply an interrupt vector to the on-board CPU.

Interrupt Sources

Twenty-six potential interrupt sources are routed to the slave PIC device and to the interrupt jumper matrix where the user can connect the desired interrupt sources to specific interrupt levels.

Memory Capabilities

DUAL-PORT MEMORY

The iSBC 286/12/14/16 boards feature 1, 2, or 4 megabytes of 0 wait-state, parity memory installed on the board. This memory, which is implemented using 256 Kb or 1 Mb DRAMs installed on a daughter board, is dual-ported to the on-board CPU bus and the Multibus (P1) interface. For those applications requiring more memory, the iSBC 286/12/14/16 boards also feature an iLBX and synchronous memory interface to increase physical memory capacity to 16 megabytes.

The iSBC 322/324 memory upgrade modules allow for expansion of on-board dual-port DRAM. The iSBC 322 upgrades an iSBC 286/12 to 2 MBytes. The iSBC 324 upgrades an iSBC 286/12 or iSBC 286/14 to 4 MBytes.

LOCAL MEMORY

Two, 28-pin sites are provided for installing up to 128 KB of EPROM firmware.

By installing an iSBC 341 EPROM expansion module, local memory can be increased to four sites to support up to 256 KB of EPROM. Local memory access time is selectable at one, two, or three wait-states and is a function of the device speed.

Serial I/O

A two-channel serial communications interface using Intel's 8274 Multi-Protocol Serial Controller (MPSC) is contained on the iSBC 286/12/14/16 boards. Two independent software selectable baud rate generators provide the MPSC with all common communication frequencies. The protocol (i.e. asynchronous, bisync, or SDLC/HDLC), data format, control character format, parity and baud rate are all under program control. Software interfacing to the MPSC can be via either a polled or interrupt driven routine. Channel A may be configured for an RS232C or RS422/RS449 interface; channel B is set for RS232C operation only. DMA operation for channel A is available if the optional 82258 (ADMA) is installed.

Programmable Timers

The iSBC 286/12/14/16 boards provide three independent, fully programmable 16-bit interval timers/event counters utilizing the Intel 8254 Programmable Interval Timer. Each counter is capable of operating in either BCD or binary modes. Two of these timers/counters are available to the systems designer to generate accurate time intervals under software control. Routing for the outputs of these counters is jumper selectable. The outputs may be independently routed to the 8259A Programmable Interrupt Controller or to the 8274 MPSC to count external events or provide baud rate generation. The third interval timer in the 8254 is dedicated to providing a clock for the programmable baud rate generator in the iSBC 286/12/14/16 boards' MPSC serial controller.

Line Printer Interface/Board ID

An 8255A Programmable Peripheral Interface (PPI) provides a Centronics compatible line printer interface, several on-board functions, and four non-dedicated input bits. Drivers are provided for a complete Centronics compatible line printer interface.

Software Reset

The software reset feature allows the 80286 microprocessor to return to Real Address mode operation from PVAM under software control. The system reset line (INIT*) and the dual-port memory are not affected, and all I/O context is preserved. The software reset is activated by a byte write to I/O location 00E0H. To distinguish the software reset from a true system initialization reset, a flag is provided. Another flag is provided that indicates whether the iSBC 286/12/14/16 board hardware (not the 80286 device) is currently configured for PVAM or Real Address Mode.

Front Panel Connector

A 14-pin connector (J4) is mounted on the top edge of the board and is designed to connect to the front panel and power supply of the system enclosure. Leads supported include Reset and Interrupt input lines from (conditioned) front panel switches, a Run signal to drive a front panel LED, a Power Fail Interrupt line that connects to the power supply, and extra power and ground leads to support miscellaneous front panel circuitry.

Multimaster Capabilities

The iSBC 286/12/14/16 boards provide full system bus arbitration control logic. This control logic allows up to three iSBC 286/12/14/16 boards or other bus masters to share the system bus using a serial (daisy chain) priority scheme and allows up to 16 masters to share the Multibus system bus with an external parallel priority decoder.

Memory Map

The memory map of the iSBC 286/12/14/16 board is shown in Figure 1. The memory maps for the iSBC 286/14 and iSBC 286/16 boards are similar, except the total amount of on-board DRAM memory is 2 or 4 MB, and the dual-port memory space is larger. The memory map, which shows the default configuration of the board, may be easily changed by the user to meet the needs of almost any system design. As a result, the iSBC 286/12/14/16 boards are particularly suited for complex multiple processor and/or multiple intelligent I/O board-based systems.

The memory map can be changed by moving on-board jumpers or by installing user-programmed PALs (programmable array logic devices).

Using only the jumpers on the iSBC 286/12/14/16 board, the Multibus window size can be set at 0 (no window), 64 KB, 128 KB, 256 KB, or 1 MB in real address mode. The Multibus window is normally not available in PVAM, however, a PAL may be programmed to provide this feature. Jumpers are also used to set aside a portion of the dual-port memory so that it may only be accessed by the CPU (single-ported memory). Block sizes of 64 KB, 128 KB, 256 KB, 512 KB or 1 MB may be selected. Finally, jumpers are used to select any of 6 EPROM memory

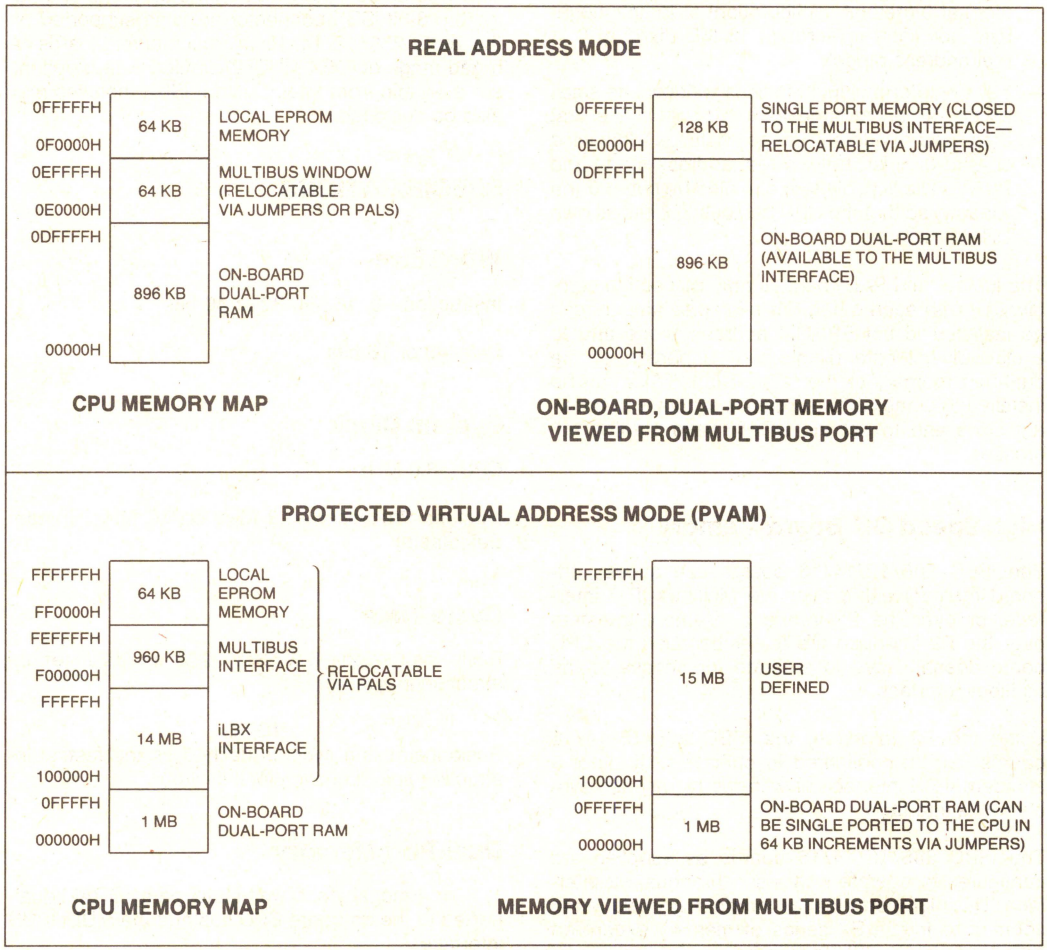


Figure 1. Memory Map for iSBC 286/12 Board (Default Configuration)

sizes ranging from 4 KB (using 2716 devices) up to 256 KB (using 27512 devices and an iSBC 341 module).

If the user needs to alter the memory map further, five PALs on the baseboard are socketed and may be replaced by custom designed devices. Using programmed PALs, the designer can:

— Set the base DRAM memory starting address (as viewed by the 80286 microprocessor) at 0 (default configuration) or to any ½ megabyte boundary up through 16 MB (0 or 512 KB in real address mode).

— Set the base DRAM memory starting address (as viewed by other boards over the Multibus interface) at 0 (default configuration) or to any megabyte boundary up through 16 MB (fixed at 0 in real address mode).

— Set single or multiple Multibus windows as small as 64 KB or as large as 1 MB within the first megabyte of address space. Multibus windowing can be enabled both in real address mode and PVAM. The window size can also be set at 0 (no window) so that the CPU can only access its own DRAM memory.

The jumper and PAL changes may be used in combination with each other. For example, jumpers can be installed to set EPROM address space and to exclusively allocate (single-port) a portion of the dual-port memory to the CPU. Then, PALs can be installed to establish two Multibus windows of different sizes and to set the DRAM base starting addresses.

High Speed Off-Board Memory

The iSBC 286/12/14/16 boards can access off-board memory either over the Multibus (P1) interface, or over the P2 interface. Memory transfers over the P2 interface are faster because the CPU board doesn't have to arbitrate for access to the Multibus interface.

Using the P2 interface, the iSBC 286/12/14/16 boards can be configured to operate with either a standard iLBX interface or with the high-speed synchronous interface.

The iSBC 286/12/14/16 boards as supplied are configured to operate with a synchronous, P2 interface. This high-performance interface is designed to connect to Intel's EX series of memory expansion boards to yield a CPU to memory read/write time of 0 wait-states. The EX memory expansion boards are available in sizes ranging from 512K bytes up to 4M bytes.

A total of four memory boards can be placed on the iLBX or synchronous interface bus. With 4M byte memory boards, this results in a total of 16M bytes on the memory expansion bus.

iSBX Bus MULTIMODULE On-Board Expansion

Two 8-, 16-bit iSBX MULTIMODULE connectors are provided on the iSBC 286/12/14/16 boards. The iSBX interface connectors on the iSBC 286/12/14/16 boards provide all signals necessary to interface to the local on-board bus, including 16 data lines for maximum data transfer rates. The iSBX MULTIMODULE Boards designed with 8-bit data paths and using the 8-bit iSBX connector are also supported on the iSBC 286/12/14/16 microcomputer boards. A broad range of iSBX MULTIMODULE Board options are available from Intel. Custom iSBX modules may also be designed.

SPECIFICATIONS

Word Size

Instruction—8, 16, 24, 32 or 40 bits

Data—8 or 16 bits

System Clock

CPU—8.0 MHz

Numeric Processor—5.3 MHz or 8.0 MHz (Jumper Selectable)

Cycle Time

Basic Instruction—8.0 MHz - 250 ns (assumes instruction in queue)

NOTE:
Basic instruction cycle is defined as the fastest instruction time (i.e. two clock cycles)

Dual-Port Memory

1, 2, or 4 megabyte, 0 wait-state, parity DRAM dual-ported to the on-board CPU bus and the MULTIBUS interface.

Local Memory

Number of sockets—two 28-pin JEDEC sites, expandable to 4 sites using iSBC 341 JEDEC Expansion Module

Maximum Size—128 KB expandable to 256 KB by installing an iSBC 341 EPROM Expansion Module. Memory size is set by jumpers on the iSBC 286/12/14/16 board.

Compatible Devices—EPROM, up to 64K x 8 (Intel 27512)

Interrupt Capacity

26 interrupt sources (total); 5 hard-wired to the 8259A PIC; 21 jumper selectable

Interrupt Levels—16 vectored requests using two 8259A devices and the 80286 microprocessor's NMI line

I/O Capability

Parallel — Line printer interface, on-board functions, and 3-bit board installed options code

Serial — Two programmable channels using one 8274 device

Timers — Three programmable timers using one 8254 device

Expansion— Two 8/16-bit iSBX MULTIMODULE connectors

Interfaces

MULTIBUS Bus—All signals TTL compatible

iSBX Bus—All signals TTL compatible

iLBX Bus—All signals TTL compatible

Synchronous Interface—All signals TTL compatible

Serial I/O— Channel A: RS232C/RS422/RS449 compatible, DCE or DTE

Channel B: RS232C compatible, DCE

Timer—All signals TTL compatible

Interrupt Requests—All TTL compatible

Serial Communications Characteristics

Synchronous—5–8 bit characters; internal or HDLC/SDLC character synchronization; automatic sync insertion; even or odd parity

Asynchronous—5–8 bit characters; break character generation; 1, 1½, or 2 stop bits; false start bit detection; even or odd parity

Physical Characteristics

Width: 12.00 in. (30.48 cm)

Height: 7.05 in. (18.00 cm)

Depth: 0.88 in. (2.24 cm)
1.16 in. (2.95 cm) with iSBX MULTIMODULE board installed

Recommended Slot spacing (without iSBX MULTIMODULE): 1.2 in. (3.0 cm)

Weight: 26 oz. (731 gm)

Electrical Characteristics

DC Power Requirements:

Maximum: +5V, 8.7A; ±12V, 35 mA (for serial I/O)

Typical: +5V, 5.7A; ±12V, 20 mA

NOTE:
Power requirements are for the default configuration. Does not include power for optional EPROM, 80287 or 82258 devices, or installed iSBX MULTIMODULE boards.

Environmental Characteristics

Operating Temperature: 0°C to 55°C with 200 LFM airflow across board (default configuration)

Relative Humidity: to 90% (without condensation)

Reference Manual

147533 — iSBC 286/12/14/16 Hardware Reference Manual (order separately)

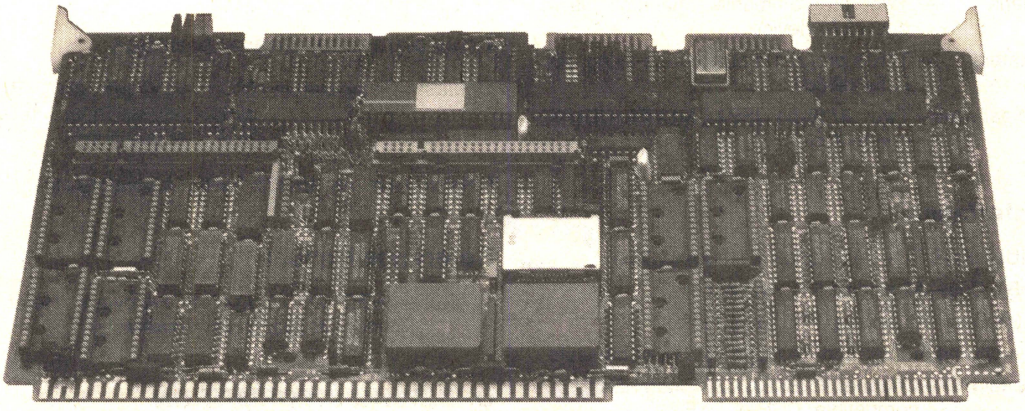
ORDERING INFORMATION

Part Number	Description
SBC 286/12	Single Board Computer with 1 MB of Memory
SBC 286/14	Single Board Computer with 2 MB of Memory
SBC 286/16	Single Board Computer with 4 MB of Memory
SBC 322	2 MB DRAM Module
SBC 324	4 MB DRAM Module

iSBC 286/10A*
SINGLE BOARD COMPUTER

- 8 MHz 80286 Microprocessor
- Supports User Installed 80287 Numeric Data Processor
- iLBX Interface for iLBX Memory Board Expansion
- 0 Wait-State Synchronous Interface to EX Memory Expansion Boards
- Eight JEDEC 28-Pin Sites for Optional SRAM/iRAM/EPROM/E^{2}PROM Components
- Optional Expansion to Sixteen JEDEC 28-Pin Sites with Two iSBC 341 Boards
- Maximum On-Board Memory Capacity 384 KB
- Two iSBX Bus Interface Connectors for I/O Expansion
- 16 Levels of Vectored Interrupt Control
- Centronics-Compatible Parallel I/O Printer Interface
- Two Programmable Multiprotocol Synchronous/Asynchronous Serial Interfaces; One RS232C, the Other RS232C or RS422/449 Compatible

The iSBC 286/10A Single Board Computer is a member of Intel's complete line of microcomputer modules and systems which take advantage of Intel's VLSI technology to provide economical, off-the-shelf, computer-based solutions for OEM applications. The CPU, system clock, memory sockets, I/O ports and drivers, serial communications interface, priority interrupt logic and programmable timers all reside on the board. The iSBC 286/10A supports both the standard iLBX interface and the synchronous high speed interface (EX-series memory boards).



280079-1

*The iSBC 286/10A is also manufactured under product code pSBC28610A by Intel Puerto Rico, Inc.

October 1993
Order Number: 280079-006

FUNCTIONAL DESCRIPTION

Overview

The iSBC 286/10A board utilizes the 80286 CPU within the Multibus system architecture, enhanced by the industry standard iLBX bus and a new, 0 wait-state, synchronous memory interface, to provide a high performance 16-bit solution. This board also includes on-board interrupt, memory and I/O features facilitating a complete signal board computer system.

Central Processing Unit

The central processor for the iSBC 286/10A board is the 80286 CPU operating at a 8.0 MHz clock rate. The 80286 CPU runs 8088 and 86 code at substantially higher speeds due to it's parallel chip architecture. In addition, the 80286 CPU provides on chip memory management and protection and virtual memory addressing of up to 1 gigabyte per task. Numeric processing power may be enhanced with the user installed 80287 numerics processor. The clock rates for the 80286 and the 80287 are independent with the 80287 rate jumper selectable at either 5.3 or 8.0 MHz.

Instruction Set

The 80286 instruction repertoire includes variable length instruction format (including double operand instructions), 8-bit and 16-bit signed and unsigned arithmetic operators for binary, BCD and unpacked ASCII data, and iterative word and byte string manipulation functions.

For enhanced numerics processing capability, the 80287 Numeric Data Processor extends the 80286 architecture and data set. Over 60 numeric instructions offer arithmetic, trigonometric, transcendental, logarithmic and exponential instructions. Supported data types include 16-, 32-, and 64-bit integer, 32- and 64-bit floating point, 18-digit packed BCD and 80-bit temporary.

Architectural Features

The 8086, 8088, 80186 and the 80286 microprocessor family contains the same basic set of registers, instructions, and addressing modes. The 80286 processor is upward compatible with the 8086, 8088, and 80186 CPUs.

The 80286 operates in two modes: 8086 real address mode, and protected virtual address mode. In 8086 real address mode, programs use real address with up to one megabyte of address space. Programs use virtual addresses in protected virtual address mode, also called protected mode. In protected mode, the 80286 CPU automatically maps 1 gigabyte of virtual addresses per task into a 16 megabyte real address space. This mode also provides memory protection to isolate the operating system and ensure privacy of each task's programs and data. Both modes provide the same base instruction set, registers, and addressing modes.

VECTORED INTERRUPT CONTROL

Incoming interrupts are handled by two on-board 8259A programmable interrupt controllers and by the 80286's NMI line. Interrupts originating from up to 16 sources are prioritized and then sent to the CPU as a vector address. Further interrupt capability is available through bus vectored interrupts where slave 8259A interrupt controllers are resident on separate iSBC boards and are then cascaded into the on-board interrupt control.

INTERRUPT SOURCES

Twenty-three potential interrupt sources are routed to the interrupt jumper matrix where the user can connect the desired interrupt sources to specific interrupt levels.

MEMORY CAPABILITIES

There are a total of eight 28-pin JEDEC sites on board. Four sites are for local memory and can contain up to 256K bytes of EPROM devices. The four other sites are known as the dual-port memory and may be addressed by the Multibus interface and the on-board CPU bus. Up to 128K bytes of either iRAM, SRAM, EPROM, or E^2PROM can reside in these sites. Both the local and dual-port memory can be expanded to eight sites each by using two iSBC 341 JEDEC expansion modules. In this way, smaller size memory devices can be used up to the 256KB (local) and 128KB (dual-port) memory capacities.

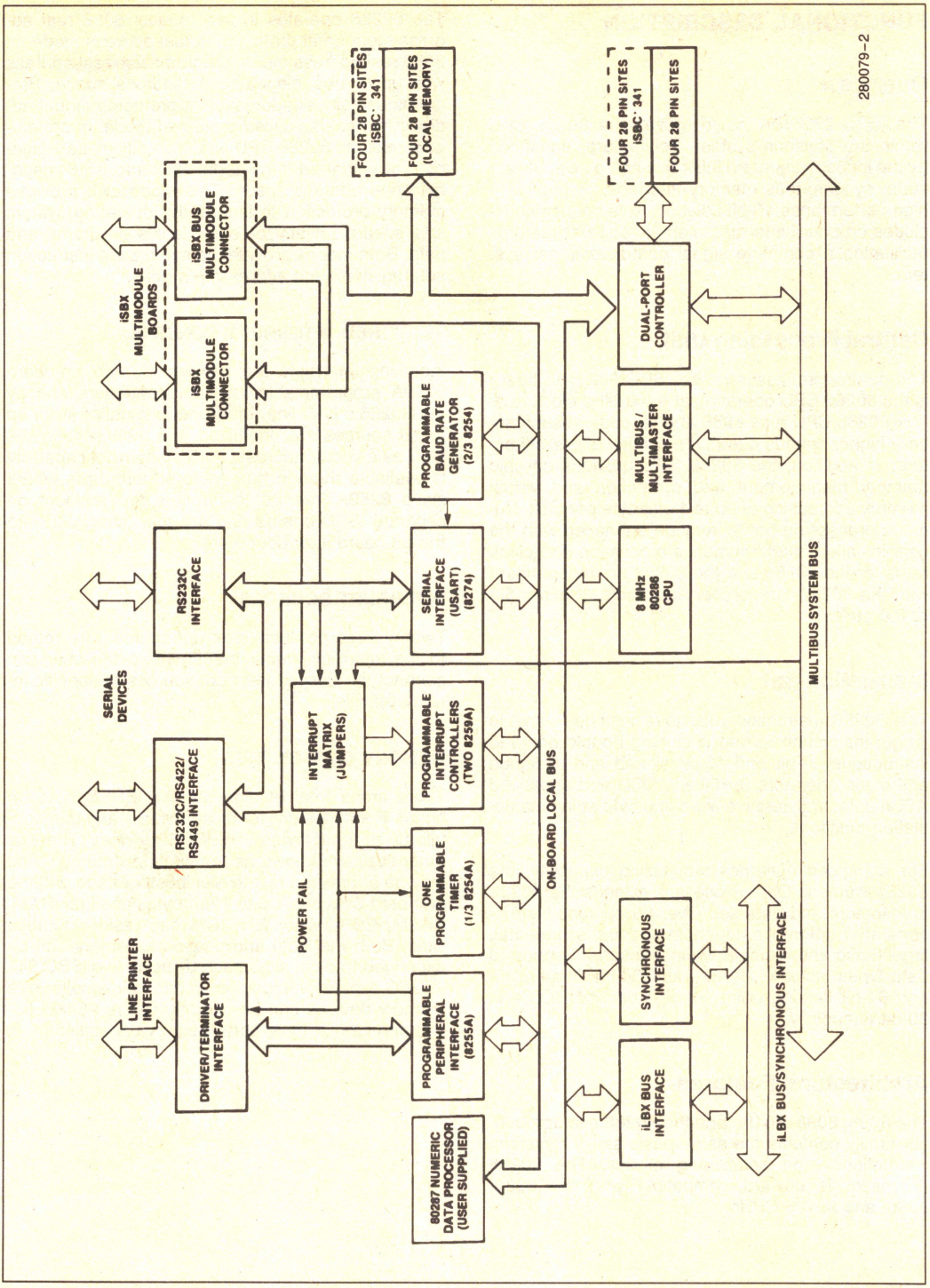


Figure 1. iSBC 286/10A Block Diagram

SERIAL I/O

A two channel serial communications interface using Intel's 8274 Multi-Protocol Serial Controller (MPSC) is contained on the iSBC 286/10A board. Two independent software selectable baud rate generators provide the MPSC with all common communication frequencies. The protocol (i.e., asynchronous, IBM* bisync, or SDLC/HDLC), data format, control character format, parity and baud rate are all under program control. Software interfacing to the MPSC can be via either a polled or interrupt driven routine. One channel may be configured for an RS232C or RS422/RS449 interface with the other channel RS232C only.

PROGRAMMABLE TIMERS

The iSBC 286/10A board provides three independent, fully programmable 16-bit interval timers/event counters utilizing the Intel 8254 Programmable Interval Timer. Each counter is capable of operating in either BCD or binary modes. Two of these timers/counters are available to the systems designer to generate accurate time intervals under software control. Routing for the outputs of these counters is jumper selectable. The outputs may be independently routed to the 8259A Programmable Interrupt Controller or to the 8274 MPSC to count external events or provide baud rate generation. The third interval timer in the 8254 is dedicated to providing a clock for the programmable baud rate generator in the iSBC 286/10A board's MPSC serial controller.

LINE PRINTER INTERFACE

An 8255A Programmable Peripheral Interface (PPI) provides a line printer interface, several on-board functions, and four non-dedicated input bits. Drivers are provided for a complete Centronics compatible line printer interface.

MULTIMASTER CAPABILITIES

The iSBC 286/10A board provides full system bus arbitration control logic. This control logic allows up to three iSBC 286/10A boards or other bus masters, to share the system bus using a serial (daisy chain) priority scheme and allows up to 16 masters to share the Multibus system bus with an external parallel priority decoder.

HIGH SPEED OFF-BOARD MEMORY

The iSBC 286/10A board can access off-board memory either over the Multibus (P1) interface, or over the P2 interface. Memory transfers over the P2 interface are faster because the CPU board doesn't have to arbitrate for access to the Multibus interface.

Using the P2 interface, the iSBC 286/10A Board can be configured to operate with either a standard iLBX interface or with a high-performance, synchronous interface.

The iSBC 286/10A Board as supplied is configured to operate with a synchronous, P2 interface. This high-performance interface is designed to connect to Intel's new EX series of memory expansion boards to yield a CPU to memory read/write time of 0 wait-states. The EX memory expansion boards are available in sizes ranging from 512K bytes up to 4M bytes and available in sizes ranging from 512K bytes up to 2M bytes. Memory expansion boards from other manufacturers that meet the iLBX standard may also be used. CPU to memory access time is usually 1 or more wait-states depending on the speed of the memory used.

A total of four memory boards can be placed on the iLBX or synchronous interface bus. With 4M byte memory boards, this results in a total of 16M bytes on the memory expansion bus.

iSBX BUS MULTIMODULE ON-BOARD EXPANSION

Two 8/16-bit iSBX MULTIMODULE connectors are provided on the iSBC 286/10A microcomputer board. The iSBX interface connectors on the iSBC 286/10A provide all signals necessary to interface to the local on-board bus, including 16 data lines for maximum data transfer rates. iSBX MULTIMODULE boards designed with 8-bit data paths and using the 8-bit iSBX connector are also supported on the iSBC 286/10A microcomputer board. A broad range of iSBX MULTIMODULE options are available from Intel. Custom iSBX modules may also be designed.

SPECIFICATIONS

Word Size

Instruction—8, 16, 24, 32 or 40 bits
Data—8 or 16 bits

System Clock

CPU—8.0 MHz
Numeric Processor—5.3 or 8.0 MHz (Jumper Selectable)

Cycle Time

Basic Instruction—8.0 MHz—375 ns; 250 ns (assumes instruction in queue)

NOTE:
Basic instruction cycle is defined as the fastest instruction time (i.e., two clock cycles)

Local Memory

Number of sockets—Four 28-pin JEDEC sites, expandable to 8 sites using iSBC 341 JEDEC Expansion Module

Maximum Size—256 KB

Compatible Devices—EPROM, up to 64K x 8 (Intel 27512)

Dual-Port Memory

Number of sockets—Four 28-pin JEDEC sites, expandable to 8 sites using iSBC 341 JEDEC Expansion Module

Maximum Size—128 KB

Compatible Devices—EPROM, up to 32K x 8 (Intel 27256)

SRAM
iRAM, up to 8K x 8 (Intel 2186)
E^2PROM, up to 2K x 8 (Intel 2817A)

I/O Capability

Parallel—Line printer interface, on-board functions, and four non-dedicated input bits

Serial—Two programmable channels using one 8274 device

Timers—Three programmable timers using one 8254 device

Expansion—Two 8/16-bit iSBX MULTIMODULE connectors

Serial Communications Characteristics

Synchronous—5–8 bit characters; internal or HDLC/SDLC character synchronization; automatic sync insertion; even or odd parity

Asynchronous—5–8 bit characters; break character generation; 1, 1½, or 2 stop bits; false start bit detection; even or odd parity

Interrupt Capacity

Potential Interrupt Sources—25, 5 fixed, 20 jumper selectable

Interrupt Levels—16 vectored requests using two 8259As and the 80286's NMI line.

INTERFACES

MULTIBUS Bus—All signals TTL compatible

iSBX Bus—All signals TTL compatible

iLBX Bus—All signals TTL compatible

Synchronous Interface—All signals TTL compatible

Serial I/O—Channel A: RS232C/RS422/RS449 compatible, DCE or DTE; Channel B; RS232C compatible, DCE only

Timer—All signals TTL compatible

Interrupt Requests—All TTL compatible

Physical Characteristics

Width: 12.00 in. (30.48 cm)
Height: 6.75 in. (17.15 cm)
Depth: 0.4 in. (1.0 cm)
Minimum Slot Spacing: 0.6 in. (1.5 cm)
Weight: 14 oz. (397 gm)

Electrical Characteristics

DC Power Requirements: +5V, 7.0A; ±12V, 50 mA (serial I/O)

NOTE:
Does not include power for optional EPROM, E^2PROM, or RAM memory devices, or installed MULTIMODULE boards

Environmental Characteristics

Operating Temperature: 0°C to 55°C with 200 LFM airflow across board

Relative Humidity: to 90% (without condensation)

Reference Manual

147532—iSBC 286/10A Hardware Reference Manual (order separately)

ORDERING INFORMATION

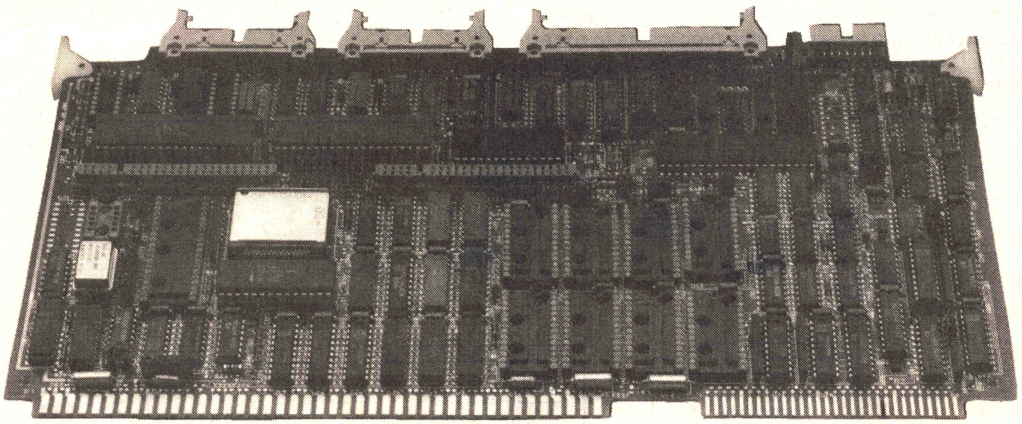
Order Code	Description
SBC 286/10A	Single Board Computer

iSBC 186/03A*
SINGLE BOARD COMPUTER

- **8.0 MHz 80186 Microprocessor with Optional 8087 Numeric Data Processor**
- **Eight (Expandable to 12) JEDEC 28-Pin Sites**
- **Six Programmable Timers and 27 Levels of Vectored Interrupt Control**
- **Multibus Interface for System Expansion and Multimaster Configuration**
- **24 Programmable I/O Lines Configurable as a SCSI Interface, Centronics Interface or General Purpose I/O**
- **Two iSBX Bus Interface Connectors for Low Cost I/O Expansion**
- **iLBX (Local Bus Extension) Interface for High-Speed Memory Expansion**
- **Two Programmable Serial Interfaces; One RS 232C, the Other RS 232C or RS 422 Compatible**

The iSBC 186/03A Single Board Computer is a member of Intel's complete line of microcomputer modules and systems that take advantage of Intel's VLSI technology to provide economical, off-the-shelf, computer-based solutions for OEM applications. The CPU, system clock, memory, sockets, I/O ports and drivers, serial communications interface, priority interrupt logic and programmable timers, all reside on the board.

The iSBC 186/03A board incorporates the 80186 CPU and SCSI interface on one board. For large memory applications, the iLBX local bus expansion maintains this high performance.



230988-1

8

*The iSBC 186/03A board is also manufactured under product code pSBC 186/03A by Intel Puerto Rico, Inc.

September 1993
Order Number: 230988-006

OVERVIEW

Operating Environment

The iSBC 186/03A single board computer features have been designed to meet the needs of numerous microcomputer applications. Typical applications include:

- Multiprocessing single board computer
- BITBUS master controller
- Stand-alone singel board system

MULTIPROCESSING SINGLE BOARD COMPUTER

Systems often need to divide system functions among multiple processors. A multiprocessing single board computer distributes an applications processing load over multiple processors that communicate over a system bus. Since these applications use the system bus for inter-processor communication, it is required that each processor has local execution memory.

The iSBC 186/03A board supports loosely coupled multiprocessing (where each processor performs a specific function) through its Multibus compatible architecture. The IEEE 796 system bus facilitates processor to processor communication, while the iLBX bus makes high-speed data and execution memory available to each CPU as shown in Figure 1. This architecture allows multiple processors to run in parallel enabling very high-performance applications.

BITBUS MASTER CONTROLLER

The BITBUS interconnect environment is a high performance low-cost microcontroller interconnect technology for distributed control of intelligent industrial machines such as robots and process controllers. The BITBUS interconnect is a special purpose serial bus which is ideally suited for the fast transmission of short messages between the microcontroller nodes in a modularly distributed system.

The iSBC 186/03A board can be implemented as the Multibus-based master controller CPU which monitors, processes and updates the control status of the distributed system. The iSBX 344 board is used to interface the iSBC 186/03A board to the BITBUS interconnect. Actual message transfer over the iSBX bus can be accomplished by either software polling by the CPU or by using the on-chip 80186 DMA hardware instead of the CPU. Using DMA, the CPU is only required to start the DMA process and then poll for the completion of the message transfer, thus dramatically improving the data transmission rate and master control processor efficiency. The maximum transfer rates over the iSBX bus for the iSBC 186/03A board are about 900 messages/second in polled mode and 2500 messages/second in DMA mode. An 8 MHz iSBC 186/03A board in DMA mode is 3 times as fast as a typical 5 MHz iSBC 86/30 board running in polled mode. The iSBC 186/03A board in DMA mode provides the highest performance/price solution for BITBUS message transmission out of all of Intel's complete line of 16-bit CPU modules.

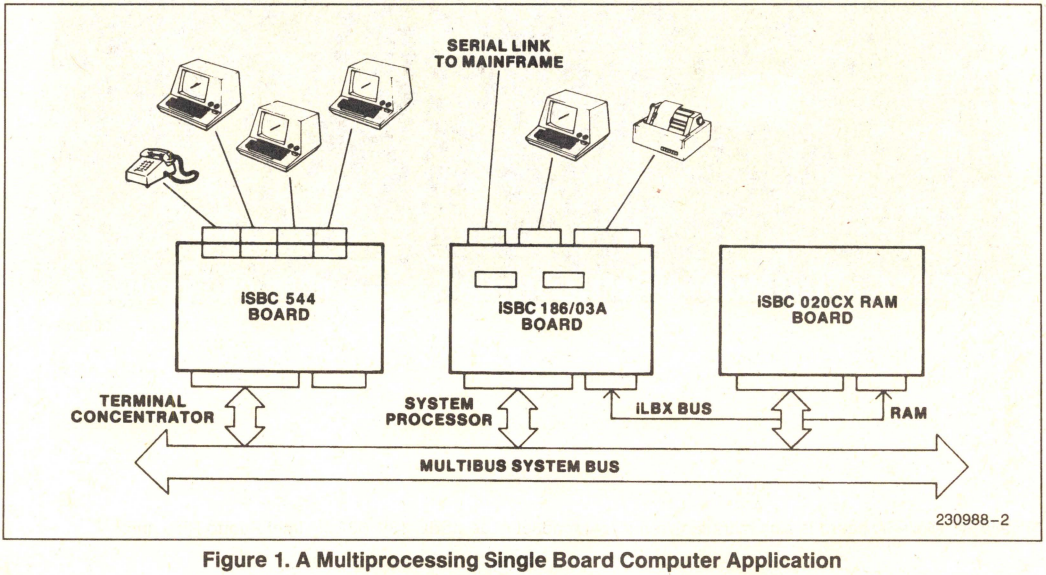


Figure 1. A Multiprocessing Single Board Computer Application

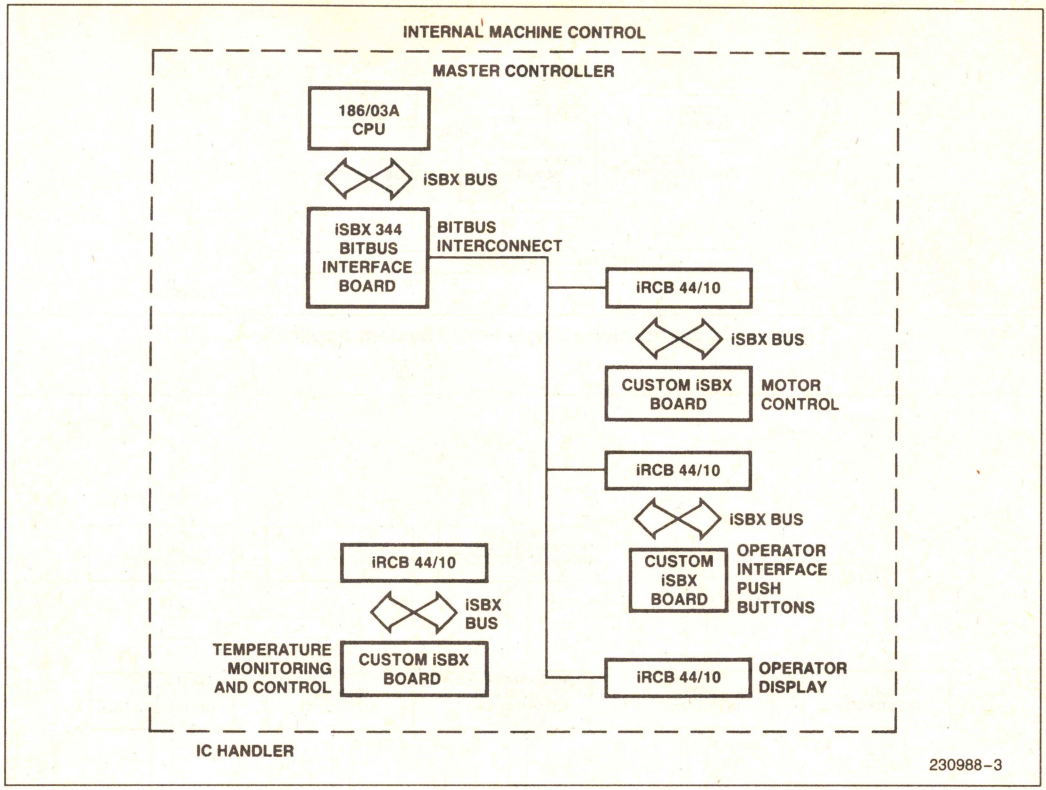


Figure 2. Sample iSBC 186/03A BITBUS Master Application

STAND-ALONE SINGLE BOARD SYSTEM

A stand-alone single board system is a complete computer system on one board. By reducing the system's board count, the single board system saves space, power, and ultimately, costs. The on-board resources need to be capable of performing all of the basic system functions. These applications typically require terminal support, peripheral control, local RAM and program execution. In previous generations of single board computers, these functions could only be obtained with multiple board solutions.

The iSBC 186/03A board integrates all the functions of a general purpose system (CPU, memory, I/O and peripheral control) onto one board. The iSBC 186/03A board can also be customized as a single board system by the selection of memory and iSBX I/O options. The board's 8 JEDEC 28-pin sockets can accommodate a wide variety of byte-wide memory devices.

If more memory is needed, an optional iSBC 341 memory site expansion board can be added to provide an additional four JEDEC sites. Two iSBX MULTIMODULE boards can be added to the iSBC 186/03A board to customize the board's I/O capabilities. As shown in Figure 3, the iSBX connectors can support a single-board system with the analog input and output modules needed by machine or process control systems.

FUNCTIONAL DESCRIPTION

8

Architecture

The iSBC 186/03A board is functionally partitioned into six major sections: central processor, memory, SCSI compatible parallel interface, serial I/O, interrupt control and Multibus bus expansion. These areas are illustrated in Figure 4.

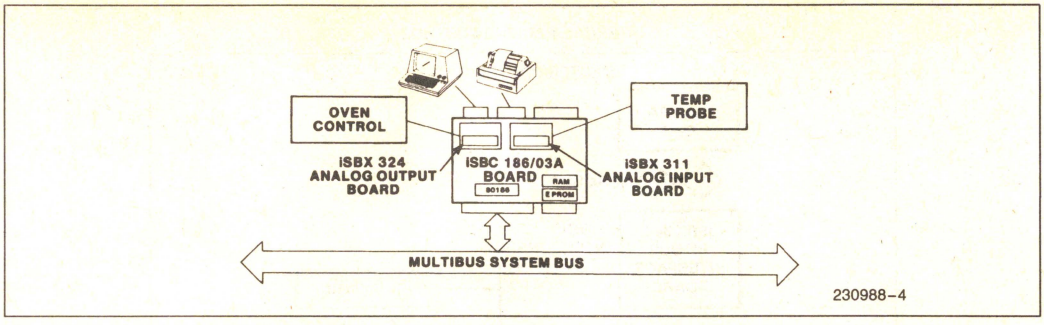


Figure 3. A Stand-Alone Single Board System Application

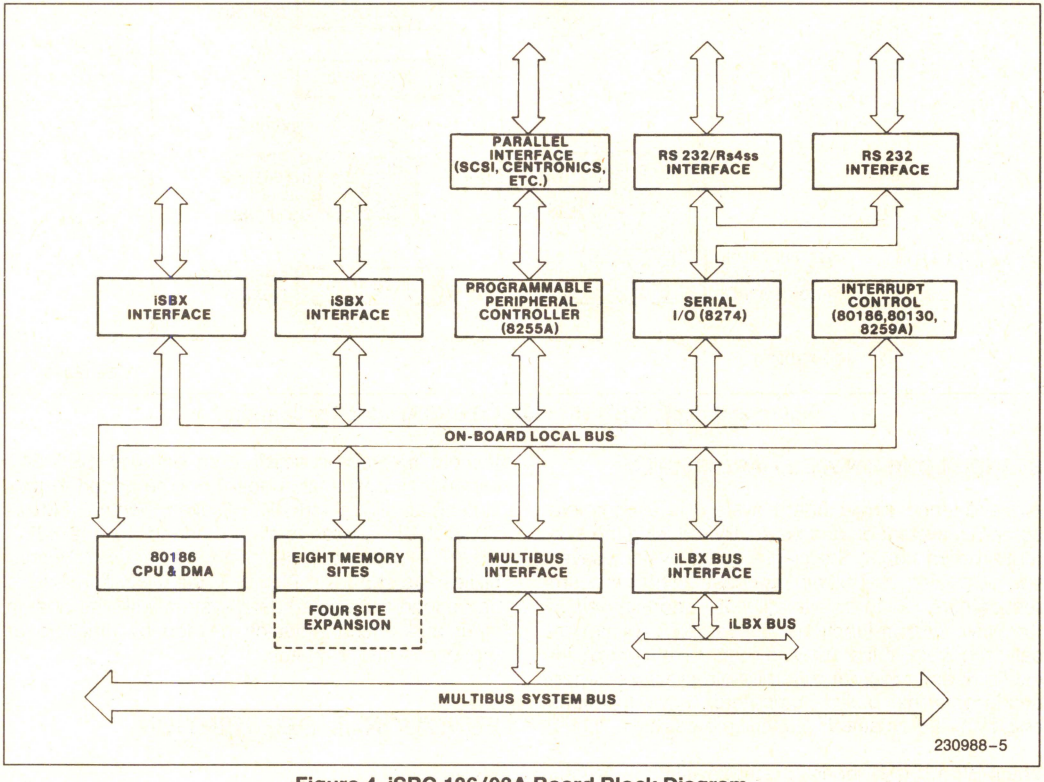


Figure 4. iSBC 186/03A Board Block Diagram

CENTRAL PROCESSOR

The 80186 component is a high-integration 16-bit microprocessor. It combines several of the most common system components onto a single chip (i.e. Direct Memory Access, Interval Timers, Clock Generator and Programmable Interrupt Controller). The 80186 instruction set is a superset of the 8086. It maintains object code compatability while adding ten new instructions. Added instructions include: Block I/O, Enter and Leave subroutines, Push Immediate, Multiply Quick, Array Bounds Checking, Shift and Rotate by Immediate, and Pop and Push All.

Use of the 80130 component is limited to the 3 timers and 8 levels of interrupts available. Direct processor execution of the 16K bytes of iRMX I Operating System nucleus primitives is not supported.

An optional 8087 Numeric Data Processor may be installed by the user to dramatically improve the 186/03A board's numerical processing power. The interface between the 8087 and 80186 is provided by the factory-installed 82188 Integrated Bus Controller which completes the 80186 numeric data processing system. The 8087 Numeric Data Processor option adds 68 floating-point instructions and eight 80-bit floating point registers to the basic iSBC 186/03A board's programming capabilities. Depending on the application, the 8087 will increase the performance of floating point calculations by 50 to 100 times.

TIMERS

The 80186 provides three internal 16-bit programmable timers. Two of these are highly flexible and are connected to four external pins (two per timer). They can be used to count external events, time external events, generate nonrepetitive waveforms, etc. As shipped on the iSBC 186/03A board, these two timers are connected to the serial interface, and provide baud rate generation. The third timer is not connected to any external pins, and is useful for real-time coding and time-delay applications. In addition, this third timer can be used as a prescaler to the other two, or as a DMA request source. The 80130 provides three more programmable timers. One is a factory default baud rate generator and outputs an 8254 compatible square wave that can be used as an alternate baud rate source to either serial channel. The 80130's second timer is used as a system timer. The third timer is reserved for use by the iRMX Operating System. The system software configures each timer independently to select the desired function. Available functions include: interrupt on terminal count, programmable one-shot, rate generator, square-wave generator, software triggered strobe, hardware triggered strobe and event counter. The contents of each counter may be read at any time during system operation.

MEMORY

There are eight JEDEC 28-pin memory sites on the iSBC 186/03A board providing flexible memory expansion. Four of these sites (EPROM sites) may be used for EPROM or E^2PROM program storage, while the other four (RAM sites) may be used for static RAM or iRAM data storage or used as additional program storage. The eight sites can be extended to twelve by the addition of an iSBC 341 MULTIMODULE board. These additional sites will provide up to 64K bytes of RAM using 8K x 8 SRAM or iRAM devices. The EPROM sites (Bank B) are compatible with 8K x 8 (2764), 16K x 8 (27128A), 32K x 8 (27256), 64K x 8 (27512) as well as 2K x 8 (2817A) and 8K x 8 (2864) E^2PROMs. The RAM sites (Bank A) are compatible with all bytewide SRAM, iRAM or NVRAM devices. NVRAM usage requires additional circuitry in order to guarantee data retention. Bank A can be reassigned to upper memory just below the assigned memory space for Bank B to support additional EPROM or E^2PROMs.

Memory addressing for the JEDEC sites depends on the device type selected. The four EPROM sites are top justified in the 1 MB address space and must contain the power-on instructions. The device size determines the starting address of these devices. The four RAM sites are, by default, located starting at address 0. The addressing of these sites may be relocated to upper memory (immediately below the EPROM site addresses) in applications where these sites will contain additional program storage. The optional iSBC 341 MULTIMODULE sites are addressable immediately above the RAM site addresses.

Power-fail control and auxiliary power are provided for protection of the RAM sites when used with static RAM devices. A memory protect signal is provided through an auxiliary connector (J4) which, when asserted, disables read/write access to RAM memory on the board. This input is provided for the protection of RAM contents during system power-down sequences. An auxiliary power bus is also provided to allow separate power to RAM for systems requiring battery back-up of read/write memory. Selection of this auxiliary RAM power bus is made via jumpers on the board.

SCSI PERIPHERAL INTERFACE

The iSBC 186/03A board includes a parallel peripheral interface that consists of three 8-bit parallel ports. As shipped, these ports are configured for general purpose I/O. The parallel interface may be reconfigured to be compatible with the SCSI disk interface by adding two user-supplied and programmed Programmable Array Logic (PAL) devices, moving jumpers and installing a user-supplied 74LS640-1 device. Alternatively, the parallel interface may be reconfigured as a DMA controlled Centronics compatible line printer interface by adding one PAL and changing jumpers. Refer to the iSBC 186/03A Hardware Reference Manual for PAL equations and a detailed implementation procedure.

The SCSI (Small Computer Systems Interface) interface allows up to 8 mass storage peripherals such as Winchester disk drives, floppy disk drives and tape drives to be connected directly to the iSBC 186/03A board. Intel's iSBC 186/03A board utilizes a single initiator, single target implementation of the SCSI bus specification. Bus arbitration and deselect/reselect SCSI features are not supported. Single host, multiple target configurations can be used. However, the iSBC 186/03A board will stay connected to one target until the transaction is completed before switching to the second target. The iSBC 186/03A board's SCSI interface implements a 5 megabit/second transfer rate. A sample SCSI application is shown in Figure 5.

The Centronics interface requires very little software overhead since a PAL device is used to provide necessary handshake timing. Interrupts are generated for printer fault conditions and a DMA request is issued for every character. The interface supports Centronics type printers compatible with models 702 and 737.

SERIAL I/O

The iSBC 186/03A Single Board Computer contains two programmable communications interfaces using the Intel 8274 Multi-Protocol Serial Controller (MPSC).

Two 80186 timer outputs are used as software selectable baud rate generators capable of supplying the serial channels with common communications frequencies. An 80130 baud rate timer may be jumpered to either serial port to provide higher frequency baud rates. The mode of operation (i.e., asynchronous, byte synchronous or bisynchronous protocols), data format, control character format, parity, and baud rate are all under program control. The 8274 provides full duplex, double buffered transmit and receive capability. Parity, overrun, and framing error detection are all incorporated in the MPSC. The iSBC 186/03A board supports operation in the polled, interrupt and DMA driven interfaces through jumper options. The default configuration is with channel A as RS422A/RS449, channel B as RS232C. Channel A can optionally be configured to support RS232C. Both channels are default configured as data set (DCE). Channel A can be reconfigured as data terminal (DTE) for connection to a modem-type device.

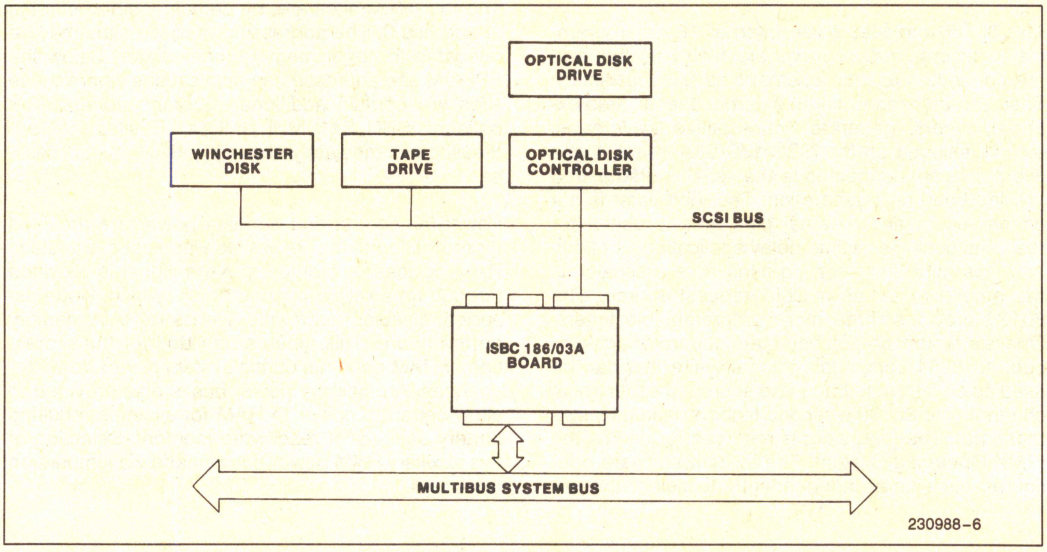


Figure 5. Sample SCSI Application

INTERRUPT CONTROL

The iSBC 186/03A board provides 27 on-board vectored interrupt levels to service interrupts generated from 33 possible sources.

The interrupts are serviced by four programmable interrupt controllers (PICs): one in the 80186 component, one in the 80130 component, one in the 8259A component and one in the 8274 component. The 80186, 8259A and 8274 PICs act as slaves to the 80130 master PIC. The highest priority interrupt is the Non-Maskable Interrupt (NMI) line which is tied directly to the 80186 CPU. This interrupt is typically used to signal catastrophic events (e.g. power failure). The PICs provide prioritization and vectoring for the other 26 interrupt requests from on-board I/O resources and from the Multibus system bus. The PICs then resolve the requests according to the programmable priority resolution mode, and if appropriate, issue an interrupt to the CPU.

MULTIBUS SYSTEM BUS—IEEE 796

The Multibus system bus is an industry standard (IEEE 796) microcomputer bus structure. Both 8- and 16-bit single board computers are supported on the IEEE 796 structure with 20 or 24 address and 16 data lines. In its simplest application, the system bus allows expansion of functions already contained on a single board computer (e.g., memory and I/O). However, the IEEE 796 bus also allows very powerful distributed processing configurations with multiple processors and intelligent slave, I/O and peripheral boards capable of solving the most demanding microcomputer applications. The Multibus system bus is supported with a broad array of board-level products, LSI interface components, detailed published specifications and application notes.

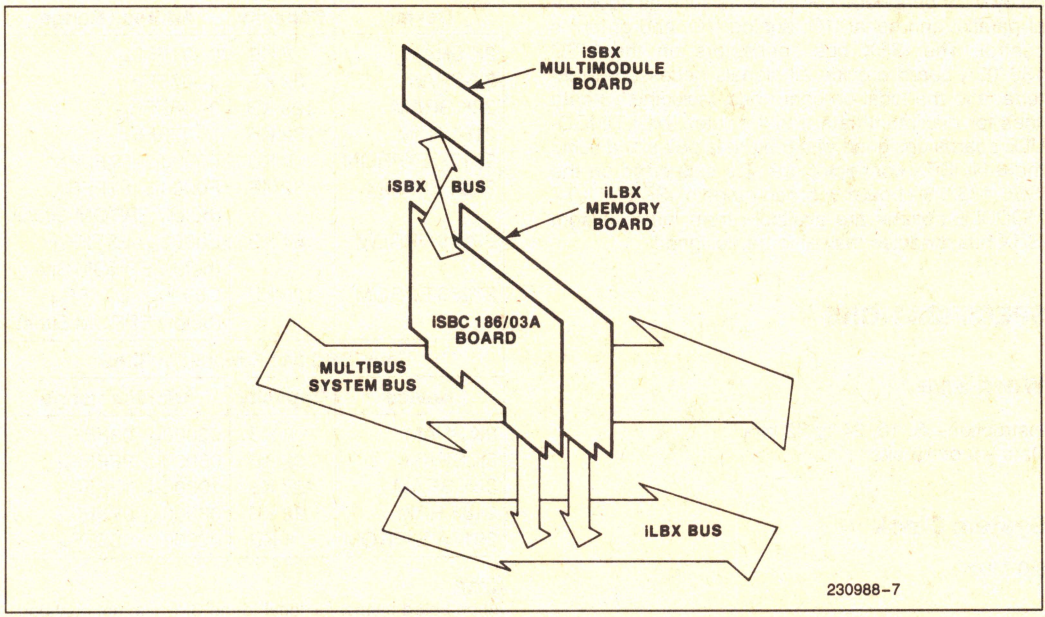


Figure 6. iSBC 186/03A Board System Architecture

iLBX BUS—LOCAL BUS EXTENSION

The iSBC 186/03A board provides a local bus extension (iLBX) interface. This standard extension allows on-board memory performance with physically off-board memory. The combination of a CPU board and iLBX memory boards is architecturally equivalent to a single board computer and thus can be called a "virtual single board computer". The iLBX bus is implemented over the P2 connector and requires independent cabling or backplane connection.

iSBX BUS MULTIMODULE ON-BOARD EXPANSION

Two iSBX MULTIMODULE board connectors are provided on the iSBC 186/03A microcomputer board. Through these connectors, additional on-board I/O functions may be added. iSBX MULTIMODULE boards optimally support functions provided by VLSI peripheral components such as additional parallel and serial I/O, analog I/O, and graphics control. The iSBX bus connectors on the iSBC 186/03A board provide all signals necessary to interface to the local on-board bus, including 16 data lines for maximum data transfer rates. MULTIMODULE boards designed with 8-bit data paths and using the 8-bit iSBX connector are also supported on the iSBC 186/03A board. A broad range of iSBX MULTIMODULE options are available from Intel. Custom iSBX bus modules may also be designed.

SPECIFICATIONS

Word Size

Instruction—8, 16, 24 or 32 bits
Data—8 or 16 bits

System Clock

8.0 MHz

Numeric Data Processor (Optional)

8087-1

Basic Instruction Cycle Time

750 ns
250 ns (assumes instruction in the queue)

NOTE:
Basic instruction cycle is defined as the fastest instruction time (i.e. two clock cycles plus instruction fetch). Zero wait-state memory is assumed.

MEMORY CAPACITY/ADDRESSING

Four EPROM Sites		
Device	**Capacity**	**Address Range**
2764 EPROM	32 KB	$F8000_H$–$FFFFF_H$
27128 EPROM	64 KB	$F0000_H$–$FFFFF_H$
27256 EPROM	128 KB	$E0000_H$–$FFFFF_H$
27512 EPROM	256 KB	$C0000_H$–$FFFFF_H$
Four RAM Sites		
Device	**Capacity**	**Address Range**
2K SRAM	8 KB	0–$01FFF_H$
8K SRAM	32 KB	0–$07FFF_H$
32K SRAM	128 KB	0–$1FFFF_H$
2186 RAM	32 KB	0–$07FFF_H$
2817A E²PROM	8 KB	$F0000_H$–$F7FFF_H$*
2764 EPROM	32 KB	$F0000_H$–$F7FFF_H$ (below EPROM Sites)
27128 EPROM	64 KB	$E0000_H$–$EFFFF_H$ (below EPROM Sites)
27256 EPROM	128 KB	$C0000_H$–$DFFFF_H$ (below EPROM Sites)
Four iSBC 341 Expansion Sites		
Device	**Capacity**	**Address Range**
2K SRAM	8 KB	02000_H–$03FFF_H$
8K SRAM	32 KB	08000_H–$0FFFF_H$
32K SRAM	128 KB	10000_H–$1FFFF_H$
2186 RAM	32 KB	08000_H–$0FFFF_H$
2817A E²PROM	8 KB	02000_H–$03FFF_H$**

NOTE:
All on board memory is local to the CPU (i.e. not dual-ported).

*Must use 8k x 8 decode option, there are four copies of the E²PROM in the 8K x 8 address area.

**(May be mixed with 2K x 8 SRAM)

Serial Communications Characteristics

Synchronous— 5–8 bit characters, internal or external character synchronization; automatic sync insertion; break character generation

Asynchronous— 5–8 bit characters; 1, ½, or 2 stop bit; false start bit detection.

Interface Compliance

MULTIBUS— IEEE 796 compliance: Master D16 M24 116 VO EL

iSBX Bus— Two 8/16 bit iSBX bus connectors allow use of up to 2 single-wide modules or 1 single-wide and 1 double-wide module. Intel iSBX bus compliance: D16/16 DMA

iLBX— Intel iLBX bus compliance: PM D16

Serial— Channel A: Configurable as RS 422A or RS 232C compatible, configurable as a data set or data terminal

Channel B: RS 232C compatible, configured as data set

Parallel I/O— SCSI (ANSI—X3T9, 2/82-s) compatible or Centronics 702 or 737 compatible (requires user supplied PALs and 74LS640-1)

ORDERING INFORMATION

Part Number	Description
SBC 186/03A	186-based Single Board Computer

REFERENCE MANUAL

iSBC 186/03A Single Board Computer Hardware Reference Manual—Order Number 148060

PHYSICAL CHARACTERISTICS

Width: 12.00 in. (30.48 cm)

Length: 7.05 in (17.90 cm)

Height: 0.50 in. (1.78 cm)

Weight: 13 ounces

ENVIRONMENTAL CHARACTERISTICS

Operating Temperature: 0°C to 60°C at 6 CFM airflow over the board.

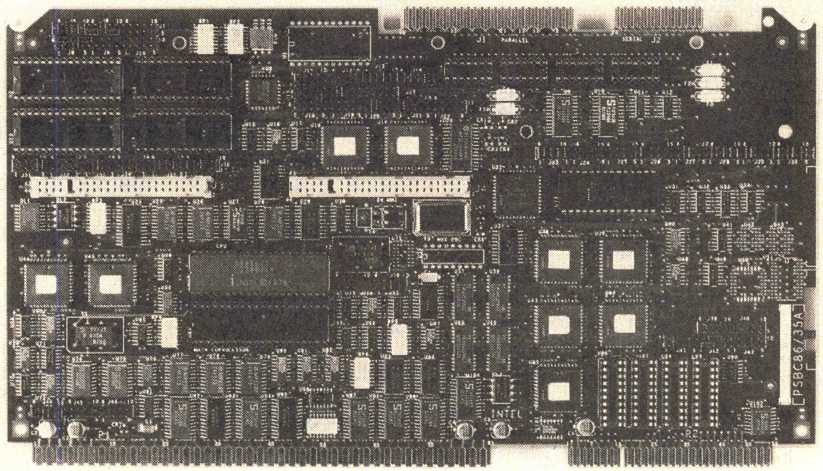
Relative Humidity: to 90% (without condensation)

ELECTRICAL CHARACTERISTICS

The maximum power required per voltage is shown below. These numbers do not include the power required by the optional memory devices, SCSI PALs, battery back-up or expansion modules.

Voltage (volts)	Max. Current (amps)	Max Power (watts)
+5	5.4	27
+12	0.04	0.48
−12	0.04	0.48

iSBC 86/35A, 86C/38, and 86/05A MULTIBUS I SINGLE BOARD COMPUTERS



281143-1

DESIGN SECURITY, LONGEVITY, AND PERFORMANCE FOR MULTIBUS I APPLICATIONS

The iSBC 86/35A MULTIBUS I CPU board combines compatibility, high reliability, increased performance, and design longevity in a single new board. This new design features the latest components to ensure the longevity of designs incorporating the iSBC 86/35A. Backwards compatibility is as easy as setting the configuration jumpers allowing the iSBC 86/35A board to function exactly as an iSBC 86/35, an 86/30, or an 86/14 MULTIBUS I CPU board.

The iSBC 86C/38 CPU board satisfies many specific low power, low heat, and high reliability requirements with a design based on CMOS technology. This board is fully compatible with iSBC 86/35.

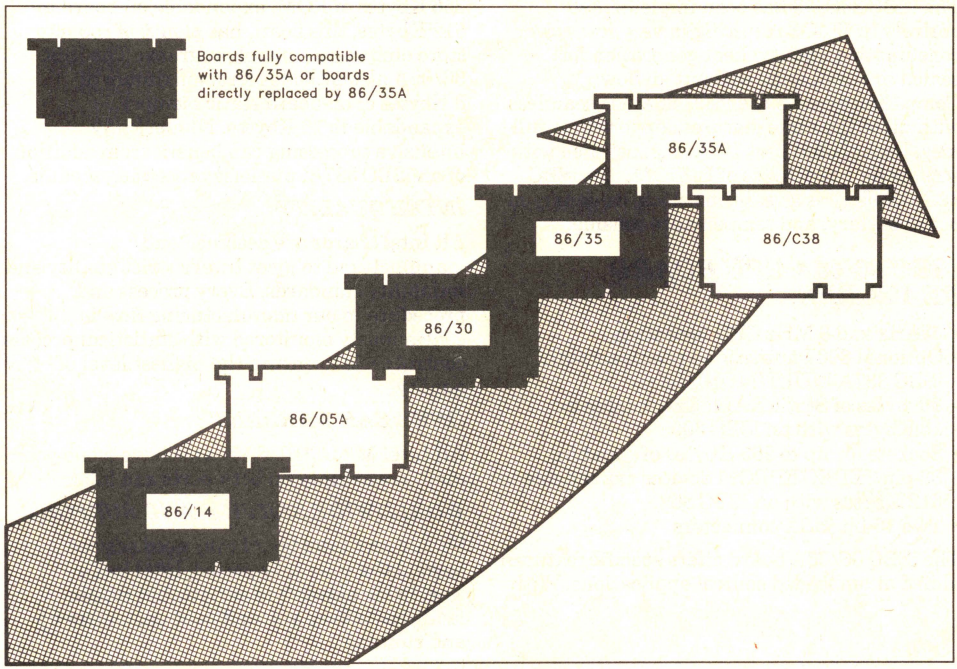
The iSBC 86/05A CPU board provides an economical solution for applications with high non-volatile memory requirements.

iSBC 86/35A CPU BOARD FEATURES

- Complete compatability with iSBC 86/30 and 86/35 boards
- 10 MHz, 8 MHz, and 5 MHz CPU operation
- 8087 numeric co-processor ready socket
- 512 Kbytes of dual port DRAM installed with sockets for expansion to 1 megabyte
- 960 Kbytes of ROM support
- Two 16-bit iSBX connectors
- I/O cycle support for 0–2 wait-states
- EPROM and DRAM support for 0–3 wait-states
- Programming support for Intel FLASH devices 28F256–28F020

October 1993
Order Number: 281143-001

FEATURES



281143-2

iSBC 86/35A CPU BOARD FEATURES

COMPATIBILITY, CONFIGURATION

The iSBC 86/35A CPU board provides full compatibility with the iSBC 86/30 and 86/35 MULTIBUS I CPU boards. Easy jumper selectable configurations enable an exact match to existing iSBC 86/35, 86/30, and 86/14 CPU boards.

10 MHz, 8 MHz, and 5 MHz CPU operation offers flexibility to match speeds to existing iSBC 86/30 and iSBC 86/35 designs. Computationally intensive applications can receive a boost to performance by plugging a 8087 math coprocessor into the available socket.

The iSBC 86/35A offers 512 Kbytes of dual-port memory. If additional memory is required, 512 Kbytes can be added to bring the on-board dual-port DRAM up to the maximum 1 megabyte.

RELIABILITY, ECONOMY, AND LONGEVITY

The extended capabilities of the new iSBC 86/35A design afford more reliability, economy, and longevity. First, the iSBC 86/35A does not require the use of separate memory or math modules. Configurations that required memory or math modules with the iSBC 86/30 or 86/35 can now run with a single card. The result is fewer boards to buy and fewer points of failure. Second, use of the latest components provides up to date technology, reliable long lived manufacturability and a design you can use with confidence for years to come.

iSBC 86C/38 CPU BOARD FEATURES

- Advanced CMOS 8 MHz 80C86 microprocessor
- 1 Mbyte of dual-port, Zero wait-state DRAM with parity
- Sockets for up to 512 Kbytes of standard 32-pin JEDEC EPROM devices
- Real-time clock/calendar with on-board battery backup
- Temperature-sensing device socket
- Optional 8087 numeric data processor with an iSBC 337A

intel

FEATURES

The iSBC 86C/38 has been implemented entirely in CMOS resulting in very low power consumption and low heat generation for sealed environments without airflow. Compatibility with the iSBC 86/35 is seamless with all of the same features supported. A full megabyte of dual port DRAM is installed with sockets for 512 Kbytes of EPROM. The iSBC 86C/38 also has a real-time clock/calendar, on-board battery, and temperature sensing.

iSBC 86/05A CPU BOARD FEATURES

- 5 MHz and 8 MHz CPU operation
- Optional 8087 numeric co-processor with iSBC 337A MULTIMODULE
- 8 Kbytes of Static RAM: Expandable to 16 Kbytes with an iSBC 302
- Sockets for up to 256 Kbytes of standard 32-pin JEDEC EPROM devices expandable to 512 Kbytes with an iSBC 302
- Two 16-bit iSBX connectors

The iSBC 86/05A board offers specific features aimed at embedded control applications. With 256 Kbytes of ROM, expandable on-board to 512 Kbytes, this board has plenty of room to store embedded control programs. The iSBC 86/05A also provides for a maximum of 8 Kbytes of on-board static memory, expandable to 16 Kbytes. Numerically intensive processing can benefit from addition of an iSBC 337A, numeric processing module.

INTEL QUALITY

All Intel boards are designed and manufactured to meet Intel's strict quality and reliability standards. Every process and procedure in our manufacturing flow is continuously monitored with statistical process control tools to ensure the highest level of quality.

ONE-YEAR WARRANTY

All Intel MULTIBUS I boards have a one-year mail-in warranty on both parts and labor.

WORLDWIDE SERVICE AND SUPPORT

Should any Intel board ever need service beyond the warranty period, Intel has a worldwide network of service and repair facilities to keep you and your customers up and running.

SPECIFICATIONS

SPECIFICATIONS iSBC 86/35A CPU BOARD

ON-BOARD RAM

512 Kbytes address range $0–7FFFF_H$ with 4 Kbytes resolution. Additional user installable 512 Kbytes requiring four 256K x 4 80 ns DRAMs.

ENVIRONMENTAL CHARACTERISTICS

Operating Temperature: 0°C to 55°C

Relative Humidity: to 90% (without condensation)

ELECTRICAL CHARACTERISTICS

DC POWER REQUIREMENTS

Configuration	Current Requirements (All Voltages ±5%)		
	+5V	+12V	−12V
Without EPROM	4.7A	25 mA	23 mA
RAM only	400 mA	—	—
With 960 Kbytes EPROM (using 27C040)	4.8A	25 mA	23 mA

System Clock

10 MHz, 8 MHz, and 5 MHz ±1% (jumper selectable)

SPECIFICATIONS

I/O CYCLE TIME

0 to 2 wait-states (jumper selectable)

Interrupt acknowledges 2 or 3 wait-states (jumper selectable)

MEMORY CAPACITY/ADDRESSING

ON-BOARD EPROM

Device Type	Number of Sites	Capacity Kbytes	Address Range
2716	4	16	FC000–FFFFF$_H$
2732	4	16	FC000–FFFFF$_H$
27C64	4	32	F8000–FFFFF$_H$
27C128	4	64	F0000–FFFFF$_H$
27C256	4	128	E0000–FFFFF$_H$
27C512	4	256	C0000–FFFFF$_H$
27C010	2	256	C0000–FFFFF$_H$
27C020	2	512	80000–FFFFF$_H$
27C040	2	960	10000–FFFFF$_H$

SPECIFICATIONS iSBC 86C/38 CPU BOARD

POWER REQUIREMENTS/ CONSUMPTION

Maximum	8 MHz		1 MHz Power Saver Mode	
+5V	1.56A	7.8W	0.8A	4.0W
+12V	0.06A	0.72W	0.06A	0.72W
−12V	0.08A	0.96W	0.08A	0.96W
Typical				
+5V	0.82A	4.1W	0.7A	3.5W
+12V	0.04A	0.48W	0.04A	0.48W
−12V	0.06A	0.72W	0.06A	0.72W

ENVIRONMENTAL REQUIREMENTS

Operating Temperature: 0°C to +60°C at **ZERO** LFM air flow

Relative Humidity: 0% to 95% non-condensing

Storage Temperature: −40°C to +70°C

SPECIFICATIONS iSBC 86/05A CPU BOARD

ON-BOARD STATIC RAM

8 Kbytes 0–1FFF$_H$

16 Kbytes 0–3FFF$_H$ (with iSBC 302 MULTIMODULE)

MEMORY CAPACITY/ADDRESSING

Device	Total Capacity Kbytes	Address Range
JEDEC 24/28 Pin Sites		
2K x 8	8	FE000–FFFFF$_H$
4K x 8	16	FC000–FFFFF$_H$
8K x 8	32	F8000–FFFFF$_H$
16K x 8	64	F0000–FFFFF$_H$
32K x 8	128	E0000–FFFFF$_H$
64K x 8	256	C0000–FFFFF$_H$
With iSBC 341 MULTIMODULE EPROM/SRAM		
2K x 8	16	FC000–FFFFF$_H$
4K x 8	32	F8000–FFFFF$_H$
8K x 8	64	F0000–FFFFF$_H$
16K x 8	128	E0000–FFFFF$_H$
32K x 8	256	C0000–FFFFF$_H$
64K x 8	512	80000–FFFFF$_H$

COMMON SPECIFICATIONS

SYSTEM CLOCK (iSBC 86C/38 and iSBC 86/05A)

8 MHz or 5 MHz ±0.1% (jumper selectable)

I/O CAPACITY

Parallel: 24 programmable lines using one 82C55A or 8255A

Serial: 1 programmable line using one 8251A iSBX MULTIMODULE: 2 iSBX boards

INTERRUPT LEVELS

Nine vectored interrupt levels expandable to 65 levels

intel

SPECIFICATIONS

SERIAL COMMUNICATIONS CHARACTERISTICS

Synchronous: 5 bit–8 bit characters; internal or external character synchronization; automatic sync insertion

Asynchronous: 5 bit–8 bit characters; break character generation: 1 bit, 1½ bits, or 2 bits; false start bit direction

INTERFACES

MULTIBUS: All signals TTL compatible

iSBX Bus: All signals TTL compatible

Parallel I/O: All signals TTL compatible

Serial I/O: RS232C compatible, configurable as a data set or data terminal

Timer: All signals TTL compatible

Interrupt Requests: All TTL compatible

PHYSICAL CHARACTERISTICS

Width: 12.00 in. (30.48 cm)

Height: 6.75 in. (17.15 cm)

Depth: 0.70 in. (1.78 cm)

Weight: 14 oz. (388 gm)

REFERENCE MANUAL

613602 — iSBC 86/35A Hardware Reference Manual

147162 — iSBC 86/05A Hardware Reference Manual

454554 — iSBC 86C/38 Single Board Computer User's Guide

(Manuals are NOT SUPPLIED they must be ordered separately)

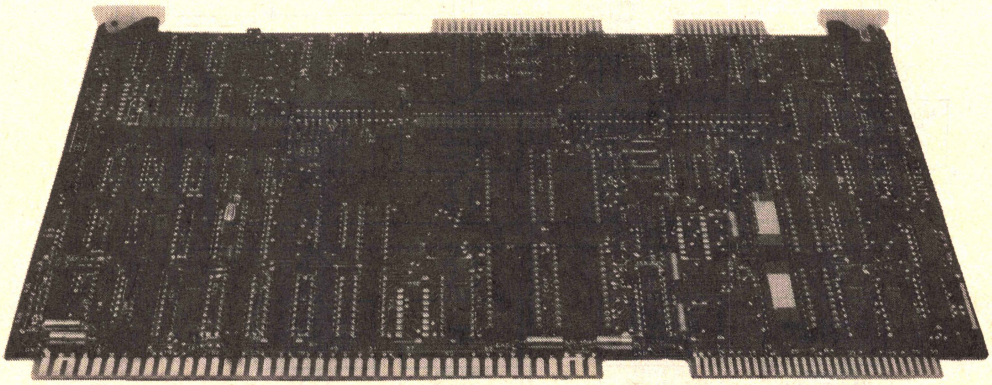
ORDER INFORMATION

Part Number	Description
SBC 86/35A	Single Board Computer
SBC 86C/38	Single Board Computer
SBC 86/05A	Single Board Computer
SBC 337A	Math MULTIMODULE
SBC 302	RAM MULTIMODULE

iSBC 88/25*
SINGLE BOARD COMPUTER

- 8-Bit 8088 Microprocessor Operating at 5 MHz
- One Megabyte Addressing Range
- Two iSBX Bus Connectors
- Optional Numeric Data Processor with iSBC 337A MULTIMODULE
- 4K Bytes of Static RAM; Expandable On-Board to 16K Bytes
- Sockets for up to 64K Bytes of JEDEC 24/28-Pin Standard Memory Devices; Expandable On-Board to 128K Bytes
- Programmable Synchronous/ Asynchronous RS232C Compatible Serial Interface with Software Selectable Baud Rates
- 24 Programmable Parallel I/O Lines
- Two Programmable 16-Bit BCD or Binary Timers/Event Counters
- 9 Levels of Vectored Interrupt Control, Expandable to 65 Levels
- MULTIBUS Interface for Multimaster Configurations and System Expansion

The iSBC 88/25 Single Board Computer is a member of Intel's complete line of OEM microcomputer systems which use Intel's technology to provide economical, self-contained, computer-based solutions for OEM applications. The CPU, system clock, read/write memory, nonvolatile read only memory, I/O ports and drivers, serial communications interface, priority interrupt logic and programmable timers, all reside on the board. The large control storage capacity makes the iSBC 88/25 board ideally suited for control-oriented applications such as process control, instrumentation and industrial automation.



143847-1

8

The iSBC 88/25 is also manufactured under product code pSBC 88/25 by Intel Puerto Rico, Inc.

September 1993
Order Number: 143847-004

FUNCTIONAL DESCRIPTION

Central Processing Unit

The central processor for the iSBC 88/25 board is Intel's 8088 CPU operating at 5 MHz. The CPU architecture includes four 16-bit byte addressable data registers, two 16-bit memory base pointer registers and two 16-bit index registers, all accessed by a total of 24 operand addressing modes for comprehensive memory addressing and for support of the data structures required for today's structured, high level languages, as well as assembly language.

Instruction Set

The 8088 instruction repertoire includes variable length instruction format (including double operand instructions), 8-bit and 16-bit signed and unsigned arithmetic operators for binary, BCD and unpacked ASCII data, and iterative word and byte string manipulation functions.

For enhanced numerics processing capability, the iSBC 337A MULTIMODULE Numeric Data Processor extends the architecture and data set. Over 60 numeric instructions offer arithmetic, trigonometric, transcendental, logarithmic and exponential instructions. Supported data types include 16, 32, and 64-bit integer, and 32 and 64-bit floating point, 18-digit packed BCD and 80-bit temporary.

Architectural Features

A 4-byte instruction queue provides pre-fetching of sequential instructions and can reduce the 750 ns minimum instruction cycle to 250 ns for queued instructions. The stack-oriented architecture readily supports modular programming by facilitating fast, simple, inter-module communication, and other programming constructs needed for asynchronous real-time systems. The memory expansion capabilities

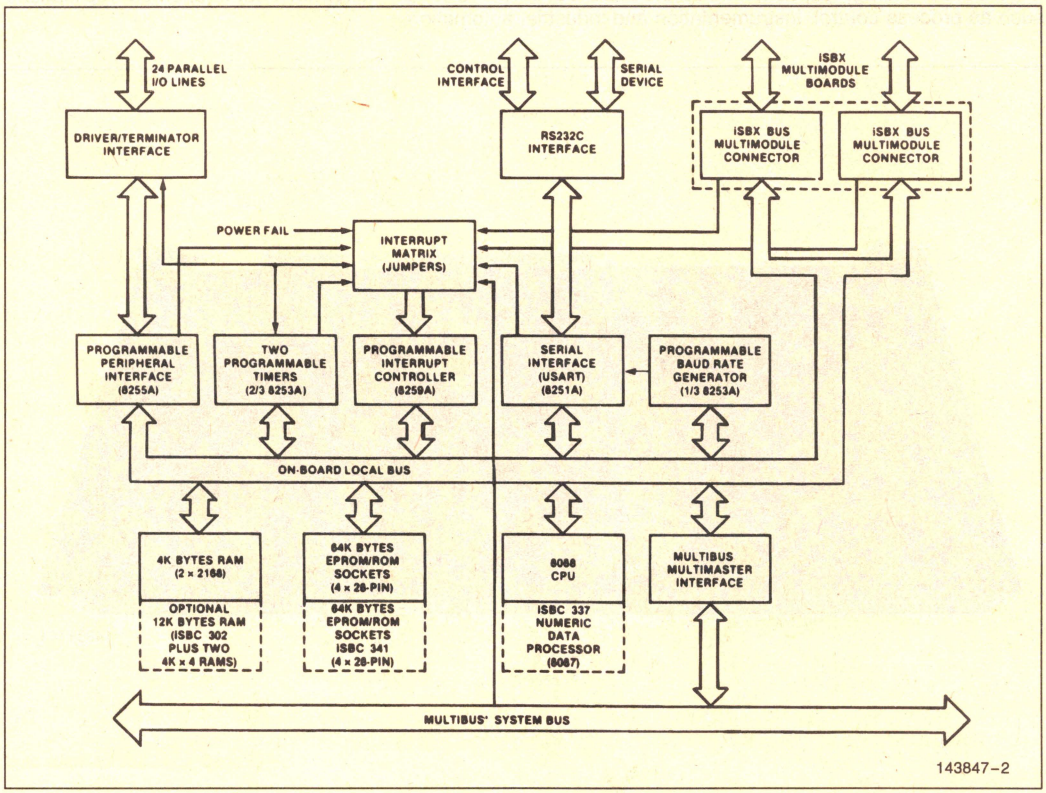


Figure 1. iSBC 88/25 Block Diagram

offer a 1 megabyte addressing range. The dynamic relocation scheme allows ease in segmentation of pure procedure and data for efficient memory utilization. Four segment registers (code, stack, data, extra) contain program loaded offset values which are used to map 16-bit addresses to 20-bit addresses. Each register maps 64 Kbytes at a time and activation of a specific register is controlled explicitly by program control and is also selected implicitly by specific functions and instructions.

Memory Configuration

The iSBC 88/25 microcomputer contains 4 Kbytes of high-speed static RAM on-board. In addition, the on-board RAM may be expanded to 12 Kbytes via the iSBC 302 8 Kbyte RAM module which mounts on the iSBC 88/25 board and then to 16 Kbytes by adding two 4K × 4 RAM devices in sockets on the iSBC 302 module. All on-board RAM is accessed by the 8088 CPU with no wait states, yielding a memory cycle time of 800 ns.

In addition to the on-board RAM, the iSBC 88/25 board has four 28-pin sockets, configured to accept JEDEC 24/28-pin standard memory devices. Up to 64 Kbytes of EPROM are supported in 16 Kbyte increments with Intel 27128 EPROMs. The iSBC 88/25 board is also compatible with the 2716, 2732 and 2764 EPROMs.

With the addition of the iSBC 341 MULTIMODULE EPROM option, the on-board capacity for these devices is doubled, providing up to 128 Kbytes of EPROM capacity on-board.

Parallel I/O Interface

The iSBC 88/25 Single Board Computer contains 24 programmable parallel I/O lines implemented using the Intel 8255A Programmable Peripheral interface. The system software is used to configure the I/O lines in any combination of unidirectional input/output and bidirectional ports indicated in Table 1. In order to take advantage of the large number of possible I/O configurations, sockets are provided for interchangeable I/O line drivers and terminators, allowing the selection of the appropriate combination of optional line drivers and terminators with the required drive/termination characteristics.

The 24 programmable I/O lines and signal ground lines are brought out to a 50-pin edge connector.

Serial I/O

A programmable communications interface using the Intel 8251A Universal Synchronous/Asynchronous Receiver/Transmitter (USART) is contained on the iSBC 88/25 board. A software selectable baud rate generator provides the USART with all common communication frequencies. The mode of operation (i.e., synchronous or asynchronous), data format, control character format, parity and baud rate are all under program control. The 8251A provides full duplex, double buffered transmit and receive capability. Parity, overrun and framing error detection are all incorporated in the USART. The RS232C compatible interface on each board, in conjunction with the USART, provides a direct interface to RS232C compatible terminals, cassettes and asynchronous and synchronous modems. The RS232C command lines, serial data lines and signal ground line are brought out to a 26-pin edge connector.

Table 1. Input/Output Port Modes of Operation

Port	Lines (qty)	Mode of Operation					Control
		Unidirectional				Bidirectional	
		Input		Output			
		Latched	Latched & Strobed	Latched	Latched & Strobed		
1	8	X	X	X	X	X	
2	8	X	X	X	X		
3	4	X		X			X(1)
	4	X		X			X(1)

NOTE:
1. Part of port 3 must be used as a control port when either port 1 or port 2 are used as a latched and strobed input or a latched and strobed output port or port 1 is used as a bidirectional port.

Programmable Timers

The iSBC 88/25 board provides three independent, fully programmable 16-bit interval timers/event counters utilizing the Intel 8253 Programmable Interval Timer. Each counter is capable of operating in either BCD or binary modes. Two of these timers/counters are available to the systems designer to generate accurate time intervals under software control. Routing for the outputs and gate/trigger inputs of two of these counters is jumper selectable.

The outputs may be independently routed to the 8259A Programmable Interrupt Controller and to the I/O terminators associated with the 8255A to allow external devices or an 8255A port to gate the timer or to count external events. The third interval timer in the 8253 provides the programmable baud rate generator for the iSBC 88/25 board RS232C USART serial port. The system software configures each timer independently to select the desired function.

iSBX MULTIMODULE On-Board Expansion

Two 8-bit iSBX MULTIMODULE connectors are provided on the iSBC 88/25 microcomputer. Through these connectors, additional on-board I/O functions such as additional parallel and serial I/O, analog I/O, mass storage device controllers and other custom interfaces may be added. By mounting directly on the single board computer, less interface logic, less power, simpler packaging, higher performance, and lower cost result when compared to other alternatives such as MULTIBUS form factor compatible boards. The iSBX connectors on the iSBC 88/25 provide all signals necessary to interface to the local on-board bus. A broad range of iSBX MULTIMODULE options are available from Intel. Custom iSBX modules may also be designed for use on the iSBC 88/25 board.

Multimaster Capabilities

For those applications requiring additional processing capacity and the benefits of multiprocessing (i.e., several CPUs and/or controllers logically sharing system tasks through communication of the system bus), the iSBC 88/25 board provides full MULTIBUS arbitration control logic. This control logic allows up to three iSBC 88/25 boards or other bus masters, including iSBC 80 and iSBC 86 family MULTIBUS compatible single board computers to share the system bus using a serial (daisy chain) priority scheme and allows up to 16 masters to share the MULTIBUS system bus with an external parallel priority decoder. In addition to the multiprocessing configurations made possible with multimaster capability, it also provides a very efficient mechanism for all forms of DMA (Direct Memory Access) transfers.

Interrupt Capability

The iSBC 88/25 board provides 9 vectored interrupt levels. The highest level is the NMI (Non-Maskable Interrupt) line which is directly tied to the 8088 CPU. This interrupt is typically used for signaling catastrophic events (e.g., power failure). The Intel 8259A Programmable Interrupt Controller (PIC) provides control and vectoring for the next eight interrupt levels. A selection of four priority processing modes is available for use in designing request processing configurations to match system requirements for efficient interrupt servicing with minimal latencies. Operating mode and priority assignments may be reconfigured dynamically via software at any time during system operation. The PIC accepts interrupt requests from all on-board I/O resources and from the MULTIBUS system bus. The PIC then resolves requests according to the selected mode and, if appropriate, issues an interrupt to the CPU.

Interrupt requests to be serviced by the iSBC 88/25 board may originate from 24 sources. All interrupt signals are brought to the interrupt jumper matrix where any combination of interrupt sources may be strapped to the desired interrupt request level on the 8259A PIC or the NMI input to the CPU directly.

Power-Fail Control and Auxiliary Power

Control logic is also included to accept a power-fail interrupt to initiate an orderly shut down of the system in the event of a power failure. Additionally, an active-low TTL compatible memory protect signal is brought out of the auxiliary connector which, when asserted, disables read/write access to RAM memory on the board. This input is provided for the protection of RAM contents during system power-down sequences. An auxiliary power bus is also provided to allow separate power to RAM for systems requiring battery backup of read/write memory. Selection of this auxiliary RAM power bus is made via jumpers on the board.

SPECIFICATIONS

Word Size

Instruction—8, 16, 24, or 32 bits
Data—8 bits

System Clock

5.00 MHz or 4.17 MHz ±0.1% (jumper selectable)

NOTE:
4.17 MHz required with the optional iSBC 337 module.

Cycle Time

BASIC INSTRUCTION CYCLE

At 5 MHz—1.2 μs
—400 ns (assumes instruction in the queue)

NOTES:
Basic instruction cycle is defined as the fastest instruction time (i.e., two clock cycles).

Memory Cycle Time

RAM—800 ns (no wait states)
EPROM—Jumper selectable from 800 ns to 1400 ns

Memory Capacity/Addressing

ON-BOARD EPROM

Device	Total Capacity	Address Range
2716	8 Kbytes	FE000-FFFFF$_H$
2732	16 Kbytes	FC000-FFFFF$_H$
2764	32 Kbytes	F8000-FFFFF$_H$
27128	64 Kbytes	F0000-FFFFF$_H$

WITH iSBC 341 MULTIMODULE EPROM

Device	Total Capacity	Address Range
2716	16 Kbytes	FC000-FFFFF$_H$
2732	32 Kbytes	F8000-FFFFF$_H$
2764	64 Kbytes	F0000-FFFFF$_H$
27128	128 Kbytes	E0000-FFFFF$_H$

ON-BOARD RAM

4 Kbytes—0-0FFF$_H$

WITH iSBC 302 MULTIMODULE RAM

12 Kbytes—0-2FFF$_H$

WITH iSBC 302 MULTIMODULE BOARD AND TWO 4K x 4 RAM CHIPS

16 Kbytes—0-3FFF$_H$

I/O Capacity

Parallel—24 programmable lines using one 8255A
Serial—1 programmable line using one 8251A
iSBX Multimodule—2 iSBX MULTIMODULE boards

Serial Communications Characteristics

Synchronous—5 8-bit characters; internal or external character synchronization; automatic sync insertion
Asynchronous—5 8-bit characters; break character generation; 1, 1½, or 2 stop bits; false start bit detection

Interfaces

Multibus: All signals TTL compatible

iSBX Bus: All signals TTL compatible

Parallel I/O: All signals TTL compatible

Serial I/O: RS232C compatible, configurable as a data set or data terminal

Timer: All signals TTL compatible

Interrupt Requests: All TTL compatible

Physical Characteristics

Width: 12.00 in. (30.48 cm)

Height: 6.75 in. (17.15 cm)

Depth: 0.70 in. (1.78 cm)

Weight: 14 oz. (388 gm)

Electrical Characteristics

DC POWER REQUIREMENTS

Configuration	Current Requirements (All Voltages ±5%)		
	+5V	+12V	−12V
Without EPROM(1)	3.8A	25 mA	23 mA
RAM only(2)	104 mA		
With 8K EPROM(3) (using 2716)	4.3A	25 mA	23 mA
With 16K EPROM(3) (using 2732)	4.4A	25 mA	23 mA
With 32K EPROM(3) (using 2764)	4.4A	25 mA	23 mA

NOTES:
1. Does not include power for optional ROM/EPROM, I/O drivers and I/O terminators.
2. RAM chips powered via auxiliary power bus in power-down mode. Does not include power for optional RAM.
3. Includes power required for 4 ROM/EPROM chips, and I/O terminators installed for 16 I/O lines; all terminator inputs low.

Environmental Characteristics

Operating Temperature: 0°C to 55°C

Relative Humidity: to 90% (without condensation)

Reference Manual

143825-001—iSBC 88/25 Hardware Reference Manual (NOT SUPPLIED)

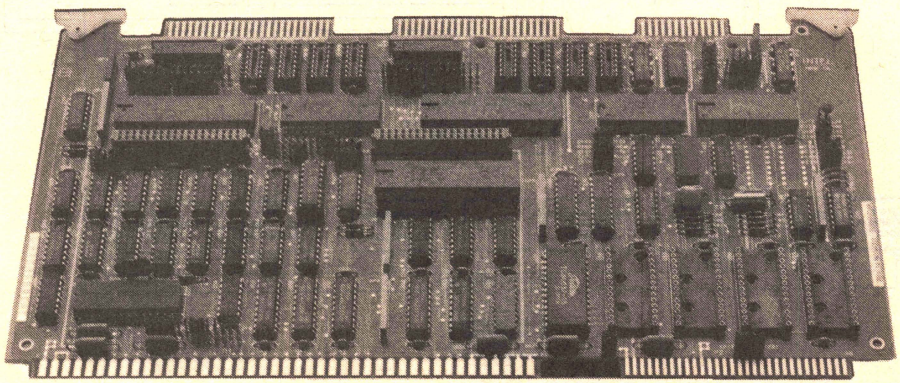
ORDERING INFORMATION

Part Number	Description
SBC 88/25	8-bit Single Board Computer with 4 Kbytes RAM

iSBC 80/24A*
SINGLE BOARD COMPUTER

- **Upward Compatible Replacement for iSBC 80/20-4 Single Board Computer**
- **8085A-2 CPU Operating at 4.8 or 2.4 MHz**
- **Two iSBX Bus Connectors for iSBX MULTIMODULE Board Expansion**
- **8K Bytes of Static Read/Write Memory**
- **Sockets for Up to 32K Bytes of Read Only Memory**
- **48 Programmable Parallel I/O Lines with Sockets for Interchangeable Line Drivers and Terminators**
- **Programmable Synchronous/ Asynchronous RS232C Compatible Serial Interface with Software Selectable Baud Rates**
- **Full Multibus Control Logic for Multimaster Configurations and System Expansion**
- **Two Programmable 16-Bit BCD or Binary Timers/Event Counters**
- **12 Levels of Programmable Interrupt Control**
- **Auxiliary Power Bus, Memory Protect, and Power-Fail Interrupt Control Logic Provided for Battery Backup RAM Requirements**

The Intel 80/24A Single Board Computer is a member of Intel's complete line of OEM microcomputer systems which use Intel's LSI technology to provide economical, self-contained computer-based solutions for OEM applications. The CPU, system clock, iSBX bus interface, read/write memory, read only memory sockets, I/O ports and drivers, serial communications interface, priority interrupt logic, and programmable timers all reside on the board. Full Multibus interface logic is included to offer compatibility with the Intel OEM Microcomputer Systems family of Single Board Computers, expansion memory options, digital and analog I/O expansion boards, and peripheral and communications controllers.



142927-1

*The iSBC 80/24A Board is also manufactured under product code pSBC 80/24A by Intel Puerto Rico, Inc.

September 1993
Order Number: 142927-005

FUNCTIONAL DESCRIPTION

Central Processing Unit

Intel's 8-bit 8085A-2 CPU is the central processor for the iSBC 80/24A board operating at either 4.8 or 2.4 MHz. The 8085A-2 CPU is directly software compatible with the Intel 8080A CPU. The 8085A-2 contains six 8-bit general purpose registers and an accumulator. The six general purpose registers may be addressed individually or in pairs, providing single and double precision operators. Minimum instruction execution time is 826 nanoseconds.

iSBX MULTIMODULE On-Board Expansion

Two 8-bit iSBX bus MULTIMODULE connectors are provided for plug-in expansion of iSBX MULTIMODULE boards.

Memory Addressing

The 8085A-2 has a 16-bit program counter which allows direct addressing of up to 64K bytes of memory. An external stack, located within any portion of read/write memory, may be used as a last-in/first-out storage area for the contents of the program counter, flags, accumulator, and all of the six general purpose registers. A 16-bit stack pointer controls the addressing of this external stack. This stack provides subroutine nesting bounded only by memory size.

Memory Capacity

The iSBC 80/24A board contains 8K bytes of static read/write memory using an 8K × 8 SRAMs. All RAM read and write operations are performed at maximum processor speed. Power for the on-board RAM may be provided on an auxiliary power bus, and memory protect logic is included for RAM battery backup requirements.

Four sockets are provided for up to 32K bytes of nonvolatile read only memory on the iSBC 80/24A board.

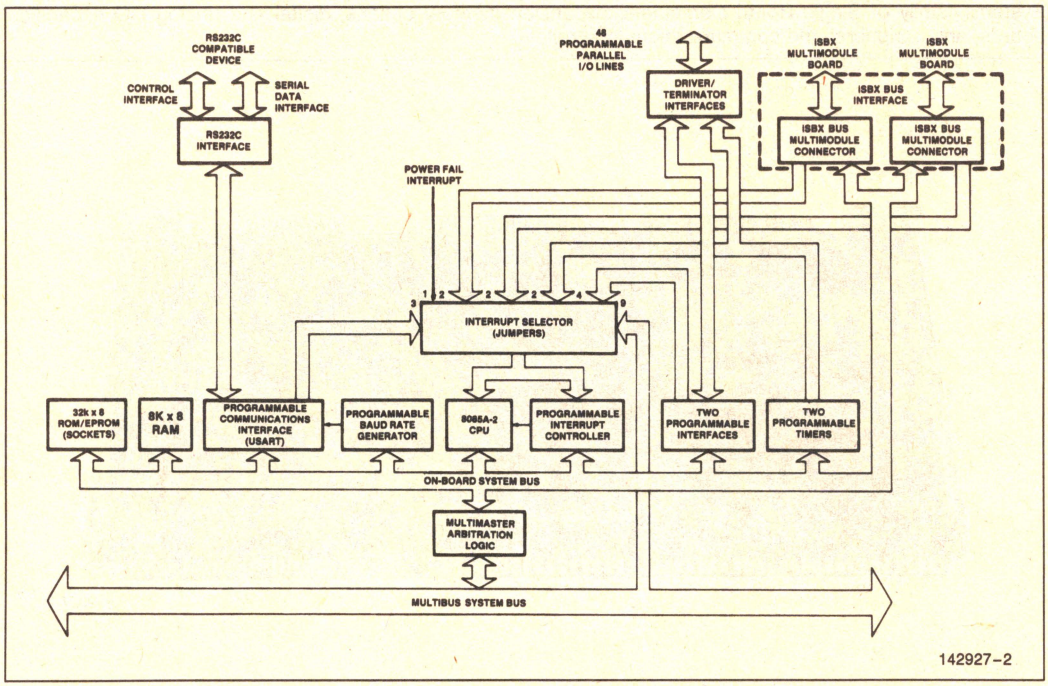


Figure 1. iSBC 80/24A Single Board Computer Block Diagram

Parallel I/O Interface

The iSBC 80/24A board contains 48 programmable parallel I/O lines implemented using two Intel 8255A Programmable Peripheral Interfaces. The system software is used to configure the I/O lines in any combination of unidirectional input/output and bidirectional ports. Therefore, the I/O interface may be customized to meet specific peripheral requirements. In order to take full advantage of the large number of possible I/O configurations, sockets are provided for interchangeable I/O line drivers and terminators. Hence, the flexibility of the I/O interface is further enhanced by the capability of selecting the appropriate combination of optional line drivers and terminators to provide the required sink current, polarity, and drive/termination characteristics for each application. The 48 programmable I/O lines and signal ground lines are brought out to two 50-pin edge connectors that mate with flat, woven, or round cables.

Serial I/O Interface

A programmable communications interface using the Intel 8251A Universal Synchronous/Asynchronous Receiver/Transmitter (USART) is contained on the iSBC 80/24A board. A software selectable baud rate generator provides the USART with all common communication frequencies. The USART can be programmed by the system software to select the desired asynchronous or synchronous serial data transmission technique (including IBM Bi-Sync). The mode of operation (i.e. synchronous or asynchronous), data format, control character format, parity, and baud rate are all under program control. The 8251A provides full duplex, double buffered transmit and receive capability. Parity, overrun, and framing error detection are all incorporated in the USART. The RS232C compatible interface, in conjunction with the USART, provides a direct interface to RS232C compatible terminals, cassettes, and asynchronous and synchronous modems. The RS232C command lines serial data lines, and signal ground line are brought out to a 26-pin edge connector that mates with RS232C compatible flat or round cable.

Multimaster Capability

For those applications requiring additional processing capacity and the benefits of multiprocessing, the iSBC 80/24A board provides full Multibus arbitration control logic. This control logic allows up to three iSBC 80/24A boards or other bus masters to share the system bus in serial (daisy chain) priority fashion, and up to 16 masters to share the Multibus system bus with the addition of an external priority network. The Multibus arbitration logic operates synchronously with a Multibus clock (provided by the iSBC 80/24A board or optionally connected directly to the Multibus clock) while data is transferred via a handshake between the master and slave modules. This allows different speed controllers to share resources on the same bus since transfers via the bus proceed asynchronously. Thus, transfer speed is dependent on transmitting and receiving devices only. This design provides slow master modules from being handicapped in their attempts to gain control of the bus, but does not restrict the speed at which faster modules can transfer data via the same bus. The most obvious applications for the master-slave capabilities of the bus are multiprocessor configurations, high speed direct memory access (DMA) operations, and high speed peripheral control, but are by no means limited to these three.

Programmable Timers

The iSBC 80/24A board provides three independent, fully programmable 16-bit interval timers/event counters utilizing the Intel 8254 Programmable Interval Timer. Each counter is capable of operating in either BCD or binary modes. Two of these timers/counters are available to the systems designer to generate accurate time intervals under software control. Routing for the outputs and gate/trigger inputs of two of these counters is jumper selectable. The outputs may be independently routed to the 8259A Programmable Interrupt Controller, to the I/O line drivers associated with the 8255A Programmable Peripheral Interface, or may be routed as inputs to the 8255A chip. The gate/trigger inputs may be routed to I/O terminators associated with the 8255A or as output connections from the 8255A. The third interval timer in the 8254 provides the programmable baud rate generator for the RS232C USART serial port. In utilizing the iSBC 80/24A board, the systems designer simply configures, via software, each timer independently to meet system requirements. Whenever a given time delay or count is needed, software commands to the programmable timers/event counters select the desired function. The contents of each counter may be read at any time during system operation with simple read operations for event counting applications, and special commands are included so that the contents of each counter can be read "on the fly".

Interrupt Capability

The iSBC 80/24A board provides vectoring for 12 interrupt levels. Four of these levels are handled directly by the interrupt processing capability of the 8085A-2 CPU and represent the four highest priority interrupts of the iSBC 80/24A board. Requests are routed to the 8085A-2 interrupt inputs—TRAP, RST 7.5, RST 6.5, and RST 5.5 (in decreasing order of priority), each of which generates a call instruction to a unique address (TRAP: 24H; RST 7.5: 3CH; RST 6.5: 34H; and RST 5.5: 2CH). An 8085A-2 JMP instruction at each of these addresses then provides linkage to interrupt service routines located independently anywhere in memory. All interrupt inputs with the exception of the trap interrupt may be masked via software. The trap interrupt should be used for conditions such as power-down sequences which require immediate attention by the 8085A-2 CPU. The Intel 8259A Programmable Interrupt Controller (PIC) provides vectoring for the next eight interrupt levels. As shown in Table 3, a selection of four priority processing modes is available to the systems designer for use in designing request processing configurations to match system requirements. Operating mode and priority assignments may be reconfigured dynamically via software at any time during system operation. The PIC accepts interrupt requests from the programmable parallel and serial I/O interfaces, the programmable timers, the system bus, iSBX bus, or directly from peripheral equipment. The PIC then determines which of the incoming requests is of the highest priority, determines whether this request is of higher priority than the level currently being serviced, and, if appropriate, issues an interrupt to the CPU. Any combination of interrupt levels may be masked, via software, by storing a single byte in the interrupt mask register of the PIC. The PIC generates a unique memory address for each interrupt level. These addresses are equally spaced at intervals of 4 or 8 (software selectable) bytes. This 32 or 64-byte block may be located to begin at any 32 or 64-byte boundary in the 65,536-byte memory space. A single 8085A-2 JMP instruction at each of these addresses then provides linkage to locate each interrupt service routine independently anywhere in memory.

Programmable Interrupt Modes

Mode	Operation
Fully nested	Interrupt request line priorities fixed at 0 as highest, 7 as lowest.
Autorotating	Equal priority. Each level, after receiving service, becomes the lowest priority level until next interrupt occurs.
Specific priority	System software assigns lowest priority level. Priority of all other levels based in sequence numerically on this assignment.
Polled	System software examines priority-encoded system interrupt status via interrupt status register.

Interrupt Request Generation

Interrupt requests may originiate from 23 sources. Two jumper selectable interrupt requests can be generated by each iSBX MULTIMODULE board. Two jumper selectable interrupt requests can be automatically generated by each programmable peripheral interface when a byte of information is ready to be transferred to the CPU (i.e., input buffer is full) or a byte of information has been transferred to a peripheral device (i.e., output buffer is empty). Three jumper selectable interrupt requests can be automatically generated by the USART when a character is ready to be transferred to the CPU (i.e., receiver channel buffer is full), a character is ready to be transmitted (i.e., the USART is ready to accept a character from the CPU), or when the transmitter is empty (i.e., the USART has no character to transmit). A jumper selectable request can be generated by each of the programmable timers. Nine interrupt request lines are available to the user for direct interface to user designated peripheral devices via the MULTIBUS system bus. A power-fail signal can also be selected as an interrupt source.

Power-Fail Control

A power-fail interrupt may be detected through the AC-low signal generated by the power supply. This signal may be configured to interrupt the 8085A-2 CPU to initiate an orderly power down instruction sequence.

SPECIFICATIONS

Word Size

Instruction— 8, 16 or 24 bits
Data — 8 bits

Cycle Time

BASIC INSTRUCTION CYCLE

826 ns (4.84 MHz operating frequency)
1.65 μs (2.42 MHz operating frequency)

NOTE:
Basic instruction cycle is defined as the fastest instruction (i.e., four clock cycles).

Memory Addressing

ON-BOARD EPROM

0-0FFF using 2708, 2758 (1 wait state)
0-1FFF using 2716 (1 wait state)
0-3FFF using 2732 (1 wait state)
using 2732A (no wait states)
0-7FFF using 2764A (no wait states)

ON-BOARD RAM

E000-FFFF

NOTE:
Default configuration—may be reconfigured to top end of any 16K boundary.

Memory Capacity

ON-BOARD EPROM

32K bytes (sockets only)

ON-BOARD RAM

8K bytes

Serial Communications Characteristics

Synchronous — 5–8 bit characters; internal or external character synchronization; automatic sync insertion

Asynchronous— 5–8 bit characters; break character generation; 1, 1½, or 2 stop bits; false start bit detectors

Interfaces

MULTIBUS	— All signals TTL compatible
iSBX Bus	— All signals TTL compatible
Parallel I/O	— All signals TTL compatible
Serial I/O	— RS232C compatible, configurable as a data set or data terminal
Timer	— All signals TTL compatible
Interrupt Requests	— All TTL compatible

System Clock (8085A-2 CPU)

4.84 or 2.42 MHz ±0.1% (jumper selectable)

Auxiliary Power

An auxiliary power bus is provided to allow separate power to RAM for systems requiring battery backup of read/write memory. Selection of this auxiliary RAM power bus is made via jumpers on the board.

Memory Protect

An active-low TTL compatible memory protect signal is brought out on the auxiliary connector which, when asserted, disables read/write access to RAM memory on the board. This input is provided for the protection of RAM contents during system power-down sequences.

Physical Characteristics

Width: 12.00 in. (30.48 cm)
Height: 6.75 in. (17.15 cm)
Depth: 0.50 in. (1.27 cm)
Weight: 12.64 oz. (354 gm)

Electrical Characteristics

DC POWER REQUIREMENTS

Configuration	Current Requirements			
	V_{CC} = +5V ±5% (max)	V_{DD} = +12V ±5% (max)	V_{BB} = −5V ±5% (max)	V_{AA} = −12V ±5% (max)
Without EPROM(1)	2.66A	40 mA	—	20 mA
RAM Only(2)	0.01A	—	—	—
With iSBC 530(3)	2.66A	140 mA	—	120 mA
With 4K EPROM(4) (using 2708)	3.28A	300 mA	180 mA	20 mA
With 4K EPROM(4) (using 2758)	3.44A	40 mA	—	20 mA
With 8K EPROM(4) (using 2716)	3.44A	40 mA	—	20 mA
With 16K EPROM(4) (using 2732A)	3.46A	40 mA	—	20 mA
With 32K EPROM(4) (using 2764A)	3.42A	40 mA	—	20 mA

NOTES:
1. Does not include power for optional EPROM, I/O drivers, and I/O terminators.
2. RAM chips powered via auxiliary power bus.
3. Does not include power for optional EPROM, I/O drivers, I/O terminators. Power for iSBC 530 Adapter is supplied via serial port connector.
4. Includes power required for four EPROM chips, and I/O terminators installed for 16 I/O lines; all terminators inputs low.

Environmental Characteristics

Operating Temperature: 0°C to 55°C

Reference Manual

148437-001— iSBC 80/24A Single Board Computer Hardware Reference Manual (NOT SUPPLIED)

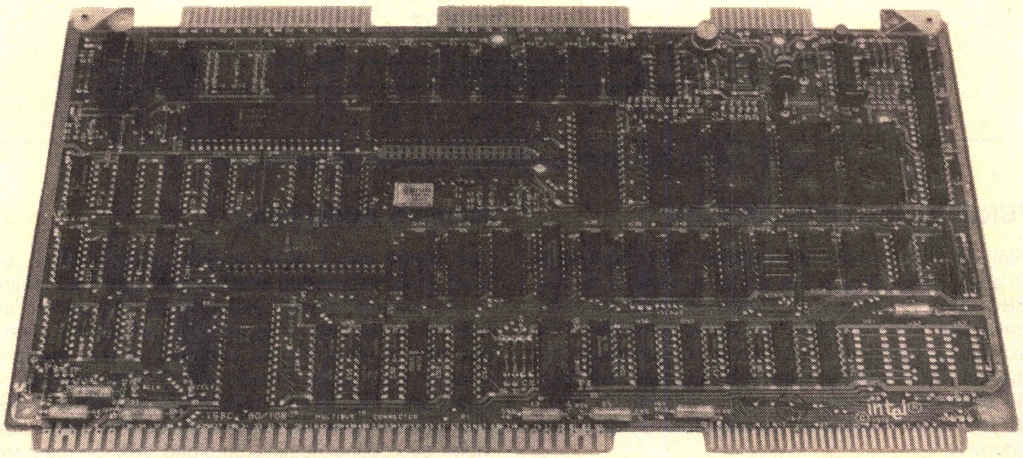
ORDERING INFORMATION

Part Number	Description
SBC 80/24A	Single Board Computer

iSBC 80/10B*
SINGLE BOARD COMPUTER

- 8080A Central Processing Unit
- One iSBX Bus Connector for iSBX MULTIMODULE Board Expansion
- 1K Byte of Read/Write Memory with Sockets for Expansion up to 4K Bytes
- Sockets for up to 16K Bytes of Read Only Memory
- 48 Programmable Parallel I/O Lines with Sockets for Interchangeable Line Drivers and Terminators
- Programmable Synchronous/ Asynchronous Communications Interface with Selectable RS232C or Teletypewriter Compatiblity
- Single Level Interrupt with 11 Interrupt Sources
- Auxiliary Power Bus and Power-Fail Interrupt Control Logic for RAM Battery Backup
- 1.04 Millisecond Interval Timer
- Limited Master MULTIBUS Interface

The Intel iSBC 80/10B board is a member of Intel's complete line of OEM microcomputer systems which use Intel's LSI technology to provide economical, self-contained computer-based solutions for OEM applications. The CPU, system clock, iSBX bus interface, read/write memory, read only memory sockets, I/O ports and drivers, serial communications interface, bus control logic, and drivers all reside on the board.



280217-1

8

*The iSBC 80/10B is also manufactured under product code pSBC 80/10B by Intel Puerto Rico, Inc.

September 1993
Order Number: 280217-003

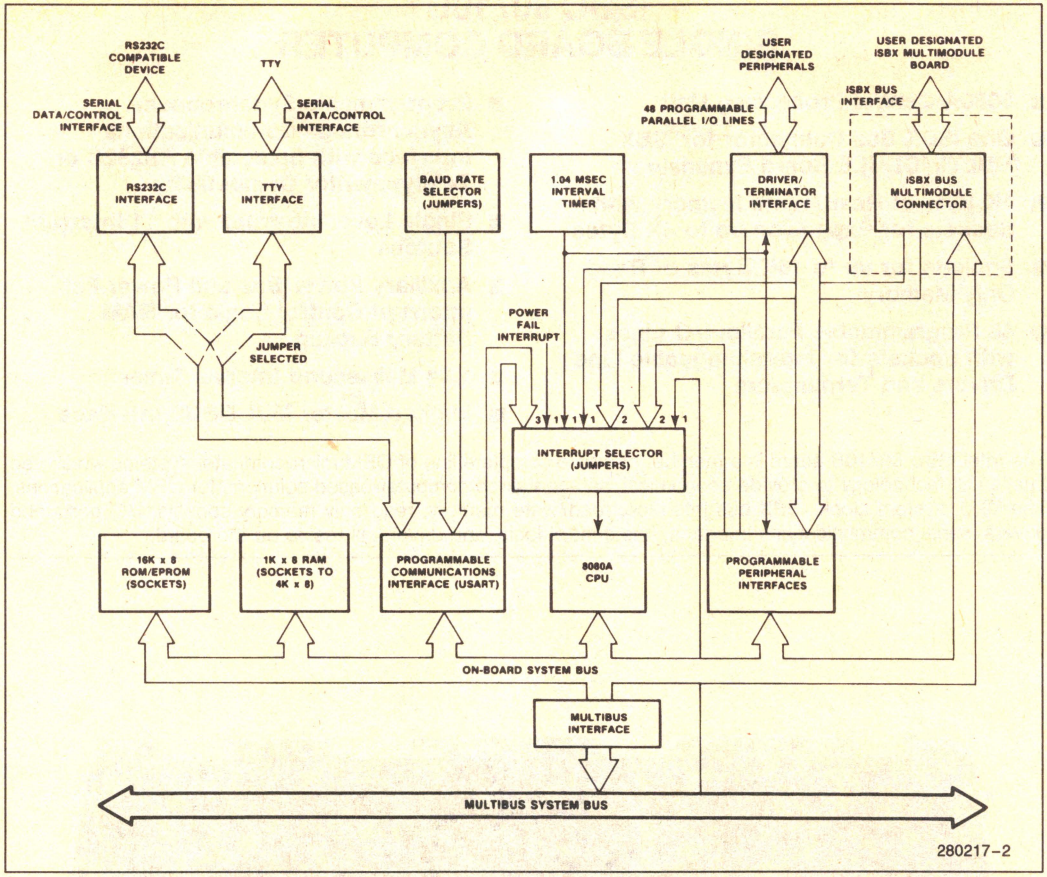


Figure 1. iSBC 80/10B Single Board Computer Block Diagram

FUNCTIONAL DESCRIPTION

Intel's 8-bit n-channel MOS 8080A CPU, fabricated on a single LSI chip, is the central processor for the iSBC 80/10B board. The 8080A contains six 8-bit general purpose registers and an accumulator. The six general purpose registers may be addressed individually or in pairs, providing both single and double precision operators. A block diagram of iSBC 80/10B board functional components is shown in Figure 1.

iSBX Bus MULTIMODULE Board Expansion

One iSBX bus connector interface is provided to accomplish plug-in expansion with any iSBX MULTIMODULE board.

Memory Addressing

The 8080A has a 16-bit program counter which allows direct addressing of up to 64K bytes of memory. An external stack, located within any portion of read/write memory, may be used as a last-in/first-out storage area for the contents of the program counter, flags, accumulator, and all of the six general purpose registers. A 16-bit stack pointer controls the addressing of this external stack. This stack provides subroutine nesting bounded only by memory size.

Memory Capacity

The iSBC 80/10B board contains 1K bytes of read/write static memory. In addition, sockets for up to 4K bytes of RAM memory are provided on board. Read/write memory may be added in 1K byte increments. Sockets for up to 16K bytes of nonvolatile read-only-memory are provided on the board. All on-board RAM, ROM or EPROM read operations are performed at maximum processor speed.

Parallel I/O Interface

The iSBC 80/10B board contains 48 programmable parallel I/O lines implemented using two Intel 8255A programmable peripheral interfaces. The system software is used to configure the I/O lines in any combination of unidirectional input/output, and bidirectional ports. In order to take full advantage of the large number of possible I/O configurations, sockets are provided for interchangeable I/O line drivers and terminators. The flexibility of the I/O interface is further enhanced by the capability of selecting the appropriate combination of optional line drivers and terminators to provide the required sink current, polarity, and drive/termination characteristics for each application. The 48 programmable I/O lines and signal ground lines are brought out to two 50-pin edge connectors that mate with flat cable or round cable.

Serial I/O Interface

A programmable communications interface using the Intel 8251A Universal Synchronous/Asynchronous Receiver/Transmitter (USART) is contained on the board. A jumper selectable baud rate generator provides the USART with all common communications frequencies. The USART can be programmed by the system software to select the desired synchronous or asynchronous serial data transmission technique (including IBM Bi-Sync). The mode of operation (i.e., synchronous or asynchronous), data format, control character format and parity are all under program control. The 8251A provides full duplex, double-buffered transmit and receive capability. Parity, overrun, and framing error detection are all incorporated in the USART. The inclusion of jumper selectable TTY or RS232C compatible interfaces on the board, in conjunction with the USART, provides a direct interface to teletypes, CRTs, RS232C compatible cassettes, and asynchronous and synchronous modems. The RS232C or TTY command lines, serial data lines, and signal ground lines are brought out to a 26-pin edge connector that mates with RS232C compatible flat or round cable.

Interrupt Capability

Interrupt requests may originate from 11 sources. Two jumper selectable interrupt requests can be automatically generated by the programmable peripheral interface when a byte of information is ready to be transferred to the CPU (i.e., input buffer is full) or a byte of information has been transferred to a peripheral device (i.e., output buffer is empty). Three jumper selectable interrupt requests can be automatically generated by the USART when a character is ready to be transferred to the CPU (i.e., receive channel buffer is full), a character is ready to be transmitted (i.e., the USART is ready to accept a character from the CPU), or when the transmitter is empty (i.e., the USART has no character to transmit). These five interrupt request lines are all maskable under program control. Two interrupt request lines may be interfaced directly to user designated peripheral devices; one via the MULTIBUS system bus and the other via the I/O edge connector. One jumper selectable interrupt request may be interfaced to the power-fail interrupt control logic. One jumper selectable interrupt request may originate from the interval timer. Two general purpose interrupt requests are jumper selectable from the iSBX interface. These two signals permit a user installed MULTIMODULE board to interrupt to 8080A CPU. The eleven interrupt request lines share a single CPU interrupt level. When an interrupt request is recognized, a restart instruction (RESTART 7) is generated. The processor responds by suspending program execution and executing a user defined interrupt service routine.

Power-Fail Control

A power-fail interrupt may be detected through the AC-low signal generated by the power supply. This signal may be configured to interrupt the 8080A CPU to initiate an orderly power down instruction sequence.

Interval Timer

A 1.04 millisecond timer is available for interval interrupts or as a clock output to the parallel I/O connector. The timer output is jumper selectable to the programmable parallel interface, the parallel I/O connector (J1), or directly to the 8080A CPU.

SPECIFICATIONS

Word Size

Instruction: 8, 16, or 24 bits
Data: 8 bits

Cycle Time

Basic Instruction Cycle: 1.95 μs

NOTE:
Basic instruction cycle is defined as the fastest instruction (i.e., four clock cycles).

Memory Addressing

On-Board ROM/EPROM
0–0FFF using 2708, 2758
0–1FFF using 2716
0–3FFF using 2732

On-Board RAM
3C00–3FFF with no RAM expansion
3000–3FFF with 2114A-5 expansion

Memory Capacity

On-Board ROM/EPROM
16K bytes (sockets only)

On-Board RAM
1K byte with user expansion in 1K increments to 4K byte using Intel 2114A-5 RAMs.

Off-Board Expansion
Up to 64K bytes using user specified combinations of RAM, ROM, and EPROM.

I/O Capacity

Parallel:	48 programmable lines
Serial:	1 transmit, 1 receive
MULTIMODULE:	1 iSBX Bus MULTIMODULE Board

Serial Communications Characteristics

Synchronous:	5–8 bit characters; internal or external character synchronization; automatic sync insertion
Asynchronous:	5–8 bit characters; break character generation; 1, 1½, or 2 stop bits; false start bit detectors

Interrupts

Single-level with on-board logic that automatically vectors the processor to location 38H using a restart instruction (RESTART 7). Interrupt requests may originate from user specified I/O (2); the programmable peripheral interface (2); the iSBX MULTIMODULE board (2); the programmable communications interface (3); the power fail interrupt (1); or the interval timer (1).

Interfaces

MULTIBUS:	All signals TTL compatible
iSBX Bus:	All signals TTL compatible
Parallel I/O:	All signals TTL compatible
Serial I/O:	RS232C or a 20 mil current loop TTY interface (jumper selectable)
Interrupt Requests:	All TTL compatible (active-low)

Clocks

System Clock: 2.048 MHz ±0.1%
Interval Timer: 1.042 ms ±0.1% (959.5 Hz)

Physical Characteristics

Width: 12.00 in (30.48 cm)
Height: 6.75 in. (17.15 cm)
Depth: 0.05 in. (1.27 cm)
Weight: 14 oz. (397.3 gm)

Electrical Characteristics

DC Power Requirements

Voltage	Without EPROM(1)	With 2708 EPROM(2)	With 2758, 2716, or 2732 EPROM(3)	Power Down Requirements (RAM and Support Circuit)
V_{CC} = +5V ±5%	I_{CC} = 2.0A(4)	3.1A	3.46A	84 mA + 140 mA/K (2114A-5)
V_{DD} = +12V ±5%	I_{DD} = 150 mA	400 mA	150 mA	Not Required
V_{BB} = −5V ±5%	I_{BB} = 2 mA	200 mA	2 mA	Not Required
V_{AA} = −12V ±5%	I_{AA} = 175 mA	175 mA	175 mA	Not Required

NOTES:
1. Does not include power required for optional ROM/EPROM, I/O drivers, or I/O terminators.
2. With four Intel 2708 EPROMS and 220Ω/330Ω for terminators, installed for 48 input lines. All terminator inputs low.
3. Same as #2 except with four 2758s, 2716s, or 2732s installed.
4. I_{CC} shown without RAM supply current. For 2114-5 add 140 mA per K byte to a maximum of 560 mA.

Environmental Characteristics

Operating Temperature: 0°C to 55°C

ORDERING INFORMATION

Part Number	Description
iSBC80/10B	Single Board Computer

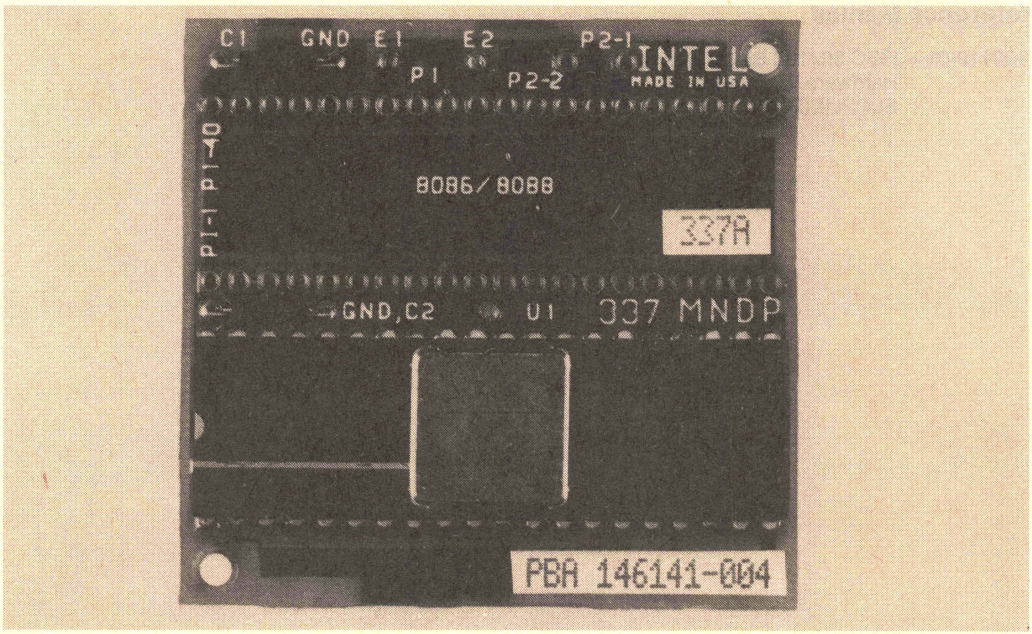
Reference Manual

9803119-01— iSBC 80/10B Single Board Computer Hardware Reference Manual (NOT SUPPLIED).

iSBC 337A MULTIMODULE NUMERIC DATA PROCESSOR

- **High Speed Fixed and Floating Point Functions for iSBC Boards**
- **Extends Host CPU Instruction Set with Arithmetic, Logarithmic, Transcendental and Trigonometric Instructions**
- **MULTIMODULE Option Containing 8087 Numeric Data Processor**
- **Supports Seven Data Types Including Single and Double Precision Integer and Floating Point**
- **Fully Supported in the Multi-Tasking Environment of the iRMX® I Operating System**

The Intel iSBC 337A MULTIMODULE Numeric Data Processor offers high performance numerics support for iSBC 86 and iSBC 88 Single Board Computer users, for applications including simulation, instrument automation, graphics, signal processing and business systems. The coprocessor interface between the 8087 and the host CPU provides a simple means of extending the instruction set with over 60 additional numeric instructions supporting seven additional data types.



280077-1

September 1993
Order Number: 280077-002

OVERVIEW

The iSBC 337A MULTIMODULE Numeric Data Processor (NDP) provides arithmetic and logical instruction extensions to the 86/88 families. The instruction set consists of arithmetic, transcendental, logical, trigonometric and exponential instructions which can all operate on seven different data types. The data types are 16-, 32-, and 64-bit integer, 32- and 64-bit floating point, 18 digit packed BCD and 80-bit temporary.

Coprocessor Interface

The coprocessor interface between the host CPU and the iSBC 337A MULTIMODULE provides easy to use and high performance math processing. Installation of the iSBC 337A is simply a matter of removing the host CPU from its socket, installing the iSBC 337A MULTIMODULE into the host's CPU socket, and reinstalling the host CPU chip into the socket provided for it on the iSBC 337A MULTIMODULE (see Figure 1).

All synchronization and timing signals are provided via the coprocessor interface with the host CPU. The two processors also share a common address/data bus. The NDP component is capable of recognizing and executing NDP numeric instructions as they are fetched by the host CPU. This interface allows concurrent processing by the host CPU and the NDP. It also allows NDP and host CPU instructions to be intermixed in any fashion to provide the maximum overlapped operation and the highest aggregate performance.

High Performance and Accuracy

The 80-bit wide internal registers and data paths contribute significantly to high performance and minimize the execution time difference between single and double precision floating point formats. This 80-bit architecture provides very high resolution and accuracy.

This precision is complemented by extensive exception detection and handling. Six different types of exceptions can be reported and handled by the NDP. The user also has control over internal precision, infinity control and rounding control.

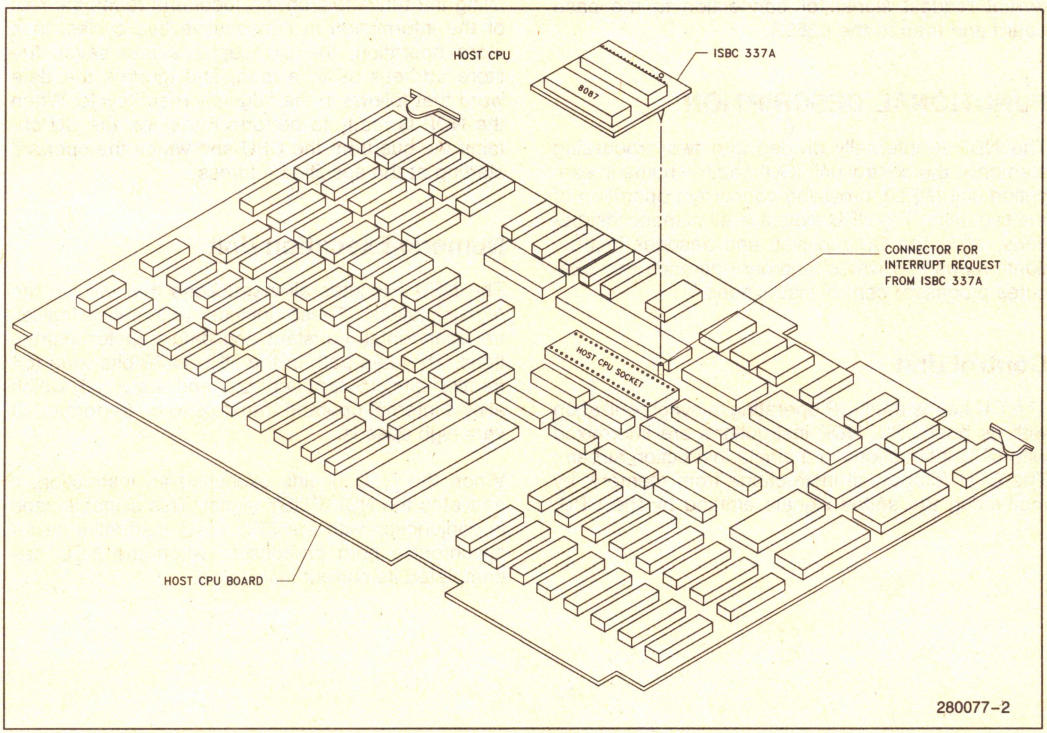


Figure 1. iSBC 337A MULTIMODULE Installation

SYSTEM CONFIGURATION

As a coprocessor to the Host CPU, the NDP is wired in parallel with the CPU. The CPU's status and queue status lines enable the NDP to monitor and decode instructions in synchronization with the CPU and without any CPU overhead. Once started, the NDP can process in parallel with and independent of the host CPU. For resynchronization, the NDP's BUSY signal informs the CPU that the NDP is executing an instruction and the CPU WAIT instruction tests this signal to insure that the NDP is ready to execute subsequent instructions.

The NDP can interrupt the CPU when it detects an error or exception. The interrupt request line is routed to the CPU through an 8259A Programmable Interrupt Controller. This interrupt request signal is brought down from the iSBC 337A MULTIMODULE to the single board computer through a single pin connector (see Figure 1). The signal is then routed to the interrupt matrix for jumper connection to the 8259A Interrupt Controller. Other iAPX designs may use a similar arrangement, or by masking off the CPU "READ" pin from the iSBC 337A socket, provisions are made to allow the now vacated pin of the host's CPU socket to be used to bring down the interrupt request signal for connection to the base board and then to the 8259A.

FUNCTIONAL DESCRIPTION

The NDP is internally divided into two processing elements, the control unit (CU) and the numeric execution unit (NEU), providing concurrent operation of the two units. The NEU executes all numeric instructions, while the CU receives and decodes instructions, reads and writes memory operands and executes processor control instructions.

Control Unit

The CU keeps the NDP operating in synchronization with its host CPU. NDP instructions are intermixed with CPU instructions in a single instruction stream. The CPU fetches all instructions from memory; by monitoring the status signals emitted by the CPU, the NDP control unit determines when a 8086-2 instruction is being fetched. The CU taps the bus in parallel with the CPU and obtains that portion of the data stream.

After decoding the instruction, the host executes all opcodes but ESCAPE (ESC), while the NDP executes only the ESCAPE class instructions. (The first five bits of all ESCAPE instructions are identical). The CPU does provide addressing for ESC instructions however.

An NDP instruction either will not reference memory, will require loading one or more operands from memory into the NDP, or will require storing one or more operands from the NDP into memory. In the first case, a non-memory reference escape is used to start NDP operation. In the last two cases, the CU makes use of a "dummy read" cycle initiated by the CPU, in which the CPU calculates the operand address and initiates a bus cycle, but does not capture the data. Instead, the CPU captures and saves the address which the CPU places on the bus. If the instruction is a load, the CU additionally captures the data word when it becomes available on the local data bus. If data required is longer than one word, the CU immediately obtains the bus from the CPU using the request/grant protocol and reads the rest of the information in consecutive bus cycles. In a store operation, the CU captures and saves the store address as in a load, and ignores the data word that follows in the "dummy read" cycle. When the NDP is ready to perform the store, the CU obtains the bus from the CPU and writes the operand starting at the specified address.

Numeric Execution Unit

The NEU executes all instructions that involve the register stack. These include arithmetic, logical, transcendental, constant and data transfer instructions. The data path in the NEU is 80 bits wide (64 fraction bits, 15 exponent bits and a sign bit) which allows internal operand transfers to be performed at very high speeds.

When the NEU begins executing an instruction, it activates the NDP BUSY signal. This signal is used in conjunction with the CPU WAIT instruction to resynchronize both processors when the NEU has completed its current instruction.

SPECIFICATIONS

Physical Characteristics

Width— 5.33 cm (2.100″)
Length— 5.08 cm (2.000″)
Height— 1.82 cm (0.718″)
iSBC 337A board + host board
Weight— 17.33 grams (0.576 oz.)

Electrical Characteristics

DC Power Requirements

V_{CC} = 5V ±5%
I_{CC} = 475 mA max.
I_{CC} = 350 mA typ.

Environmental Characteristics

Operating Temperature—0°C to 55°C with 200 linear feet/minute airflow

Relative Humidity—Up to 90% R.H. without condensation.

Reference Manual

147163-001—iSBC 337A MULTIMODULE Numeric Data Processor Hardware Reference Manual (NOT SUPPLIED WITH MULTIMODULE BOARD).

ORDERING INFORMATION

Part Number	Description
SBC 337A	MULTIMODLE Numeric Data Processor

8

Multibus I Memory Expansion Boards

9

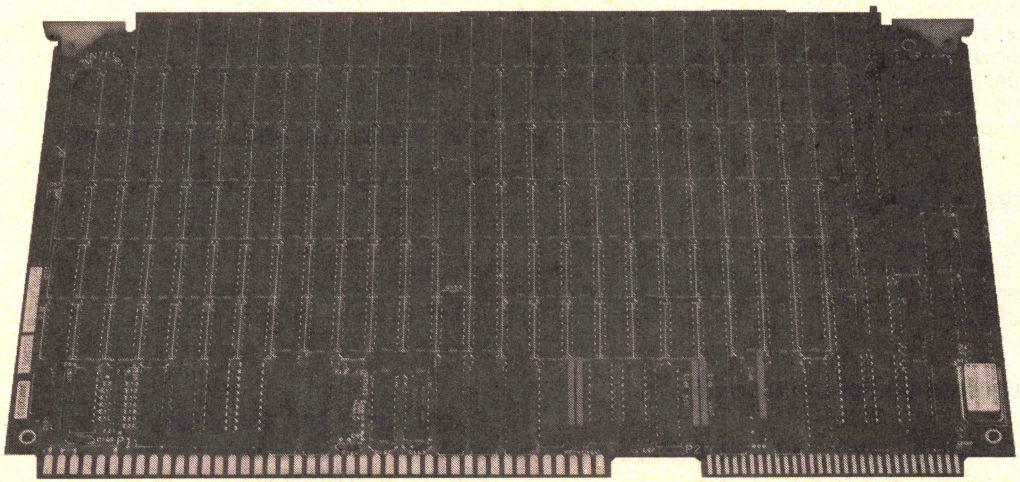
iSBC 010EX, 020EX, and 040EX* HIGH PERFORMANCE RAM BOARDS

- **0 Wait States at 8 MHz Performance with the iSBC 286/10A, iSBC 286/12 Board**
- **Dual Port Capability Via Multibus and High Speed Synchronous Interface**
- **Configurable to Function Over iLBX Bus**
- **On-Board Parity Generator/Checker**
- **Independently Selectable Starting and Ending Addresses**
- **16 Megabyte Addressing Capability**
- **1024K Byte, 2048K Byte, and 4096K Byte Densities Available**

The iSBC 010EX, iSBC 020EX, and iSBC 040EX RAM memory boards are members of Intel's complete line of iSBC memory and I/O expansion boards. The EX boards are dual ported between the Multibus interface and one of two types of dedicated memory buses. The dedicated buses are the iLBX bus and a high speed interface. The EX series of RAM-boards can be configured to be accessed over the iLBX bus, as well as Multibus bus, to provide memory support for the iSBC 286/10 board, iSBC 186/03A, or iSBC 386/12. The EX boards are default configured to run over the Multibus interface and the high speed interface. This provides 0 wait state 8 MHz memory support for the iSBC 286/10A and iSBC 286/12 boards.

The EX RAM-boards generate byte oriented parity during all write operations and perform parity checking during all read operations. An on-board LED provides a visual indication that a parity error has occurred.

The iSBC 010EX, iSBC 020EX, and iSBC 040EX boards contain 1M byte, 2M bytes, and 4M bytes of read/write memory using 256K dynamic RAM components.



280142-1

9

*The iSBC 010EX, 020EX and 040EX Boards are also manufactured under product code pSBC 010EX, 020EX and 040EX by Intel Puerto Rico, Inc.

September 1993
Order Number: 280142-003

FUNCTIONAL DESCRIPTION

General

The iSBC 010EX, 020EX, and 040EX RAM boards are physically and electrically compatible with the Multibus interface standard, IEEE-796, as outlined in the Intel Multibus architecture specification.

Dual Port Capabilities

The "EX" series of RAM-Boards can be accessed by the Multibus interface, and either the iLBX Bus, or the high speed synchronous interface (see Figures 1 and 2). The EX series require jumper and PAL configuration to be accessed over iLBX Bus.

Intel's iLBX interface is an unarbitrated bus architecture which allows direct transfer of data between the CPU and the memory boards without accessing the Multibus bus. Due to the unarbitrated nature of the iLBX interface, significant improvements in memory access times compared to the Multibus bus accesses result. The EX Boards provide 1 wait state performance at 6 MHz and 2 wait states at 8 MHz over the iLBX board. The EX Memory Board Hardware Reference Manual should be consulted for details.

The high speed synchronous interface, like the iLBX Bus, is a bus architecture which allows direct transfer of data between the CPU and the memory boards without accessing the Multibus bus. This high speed interface runs synchronously with the iSBC 286/10A and iSBC 286/12 to provide 0 wait state performance at 8 MHz.

System Memory Size

Maximum system memory size with this series of boards is 16 megabytes. Memory partitioning is independent for the Multibus interface and the iLBX interface.

Address Selection/Memory

SELECTABLE STARTING ADDRESS

A 512K boundary select is implemented on the iSBC 010EX board. A 1M boundary is implemented on the iSBC 020EX and iSBC 040EX boards.

SELECTABLE ENDING ADDRESS

The ending address is selectable as memory size minus select options of 0, 128K, 256K, or 512K on all of the EX boards.

PARITY INTERRUPT CLEAR

The I/O address of the Parity Interrupt Clear circuitry is jumperable to any one of 256 addresses.

SPECIFICATIONS

Word Size Supported

8- or 16-bits.

Memory Size

1,048,576 bytes (iSBC 010EX board)
2,097,152 bytes (iSBC 020EX board)
4,194,304 bytes (iSBC 040EX board)

Access Times (All densities)

MULTIBUS SYSTEM BUS

Read/Full Write— 375 ns (max)
Write Byte— 375 ns (max)

HIGH SPEED SYNCHRONOUS INTERFACE

Read/Full Write— 167 ns (max)
Write Byte— 132 ns (max)

iLBX BUS

Read/Full Write— 295 ns (max)
Write Byte— 116 ns (max)

Cycle Times (All densities)

MULTIBUS SYSTEM BUS

Read/Full Write— 625 ns (max)
Write Byte— 625 ns (max)

HIGH SPEED SYNCHRONOUS INTERFACE

Read/Full Write— 250 ns (max)
Write Byte — 250 ns (max)

iLBX BUS

Read/Full Write— 437.5 ns (max)
Write Byte — 437.5 ns (max)

Memory Partitioning

Maximum System memory size is 16M Bytes for the Multibus, iLBX bus and the high speed interface.

BASE ADDRESS

Board	Base Address
iSBC 010EX Board	any 512K boundary in first 8 megabytes
iSBC 020EX Board	any 1M boundary
iSBC 040EX Board	any 1M boundary

Power Requirements

Voltage—5 VDC ±5%

Product	Current
iSBC 010EX Board	3.4A (typ) 5.0A (max)
iSBC 020EX Board	3.7A (typ) 5.2A (max)
iSBC 040EX Board	3.9A (typ) 5.5A (max)

ENVIRONMENTAL REQUIREMENTS

Operating Temperature: 0°C to 60°C airflow of 5 cubic feet per minute

Storage Temperature: −40°C to +75°C

Operating Humidity: To 90% without condensation

PHYSICAL DIMENSIONS

Width: 12 inches (30.48 cm)
Height: 6.75 inches (17.15 cm)
Thickness: 0.50 inches (1.27 cm)
Weight: iSBC 010EX board: 9.0 ounces (2550 gm)
iSBC 020EX board: 13.5 ounces (3830 gm)
iSBC 040EX board: 18.0 ounces (5100 gm)

REFERENCE MANUALS

144456-001— Intel iLBX Specification

ORDERING INFORMATION

Part Number	Description
SBC010EX	1M byte RAM board with parity
SBC020EX	2M byte RAM board with parity
SBC040EX	4M byte RAM board with parity
EXASYNCX86	PALs and jumper configuration for iLBX mode

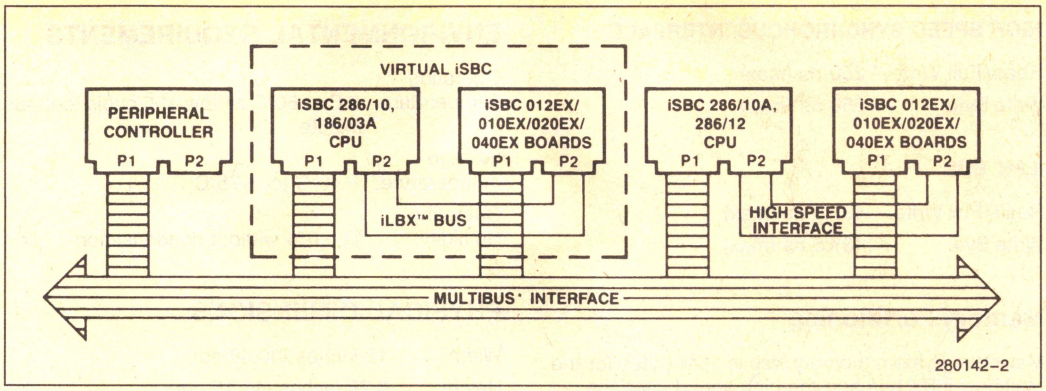


Figure 1. Typical iLBX System Configuration

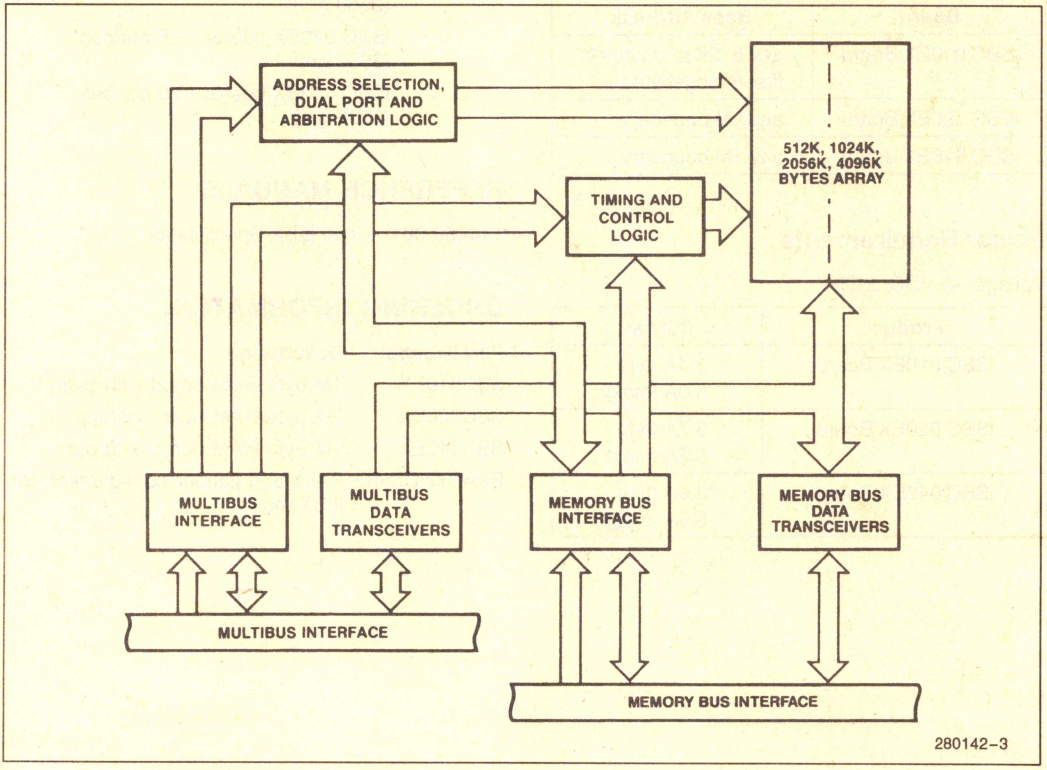


Figure 2. iSBC EX Memory Board Block Diagram

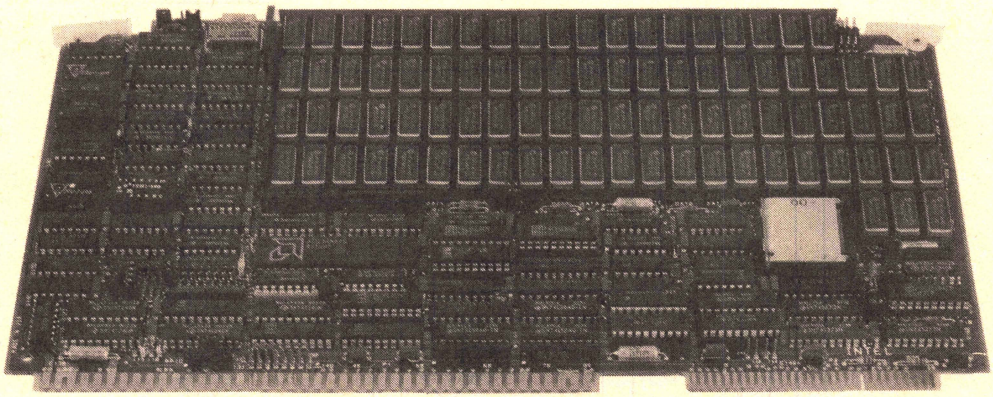
iSBC 020CX* iLBX RAM BOARD

- **Dual Port Capability via Multibus and iLBX Interfaces**
- **Single Bit Error Correction and Double Bit Error Detection Utilizing Intel 8206 ECC Device**
- **512K Byte, 1024K Byte, and 2048K Byte Versions Available**
- **Control Status Register Supports Multiple ECC Operating Modes**
- **Error Status Register Provides Error Logging by Host CPU Board**
- **16 Megabyte Addressing Capability**
- **Supports 8- or 16-bit Data Transfer and 24-bit Addressing**
- **Auxiliary Power Bus and Memory Protect Logic for Battery Back-Up RAM Requirements**

The iSBC 020CX RAM memory board is a member of Intel's complete line of iSBC memory and I/O expansion boards. The dual port feature of the CX series of RAM-boards allow access to the memory of both the Multibus and iLBX bus interfaces.

In addition to the dual port features the "CX" series of RAM-boards provide Error Checking and Corrections Circuitry (ECC) which can detect and correct single bit errors and detect, but not correct, double and most multiple bit errors.

The iSBC 020 CX board contains 2048K bytes of read/write memory using 256K dynamic RAM components.



231023-1

9

*The iSBC 020CX Board is also manufactured under product code pSBC 012CX, 010CX, and 020CX by Intel Puerto Rico, Inc.

September 1993
Order Number: 231023-003

FUNCTIONAL DESCRIPTION

General

The iSBC 020CX RAM board is physically and electrically compatible with the Multibus interface standard, IEEE-796, as outlined in the Intel Multibus specification. In addition the CX series of RAM-boards are physically and electrically compatible with the iLBX bus (Local Bus Extension) interface as outlined in the Intel iLBX Specification (see Figure 1).

Dual Port Capabilities

The "CX" series of RAM-boards can be accessed by either the Multibus interface or the iLBX interface (see Figure 2). Intel's iLBX interface is an unarbitrated bus architecture which allows direct transfer of data between the CPU and the memory boards without accessing the Multibus bus. Due to the unarbitrated nature of the iLBX interface significant improvements in memory access times result, typically a 2-6 Wait State improvement over Multibus memory access.

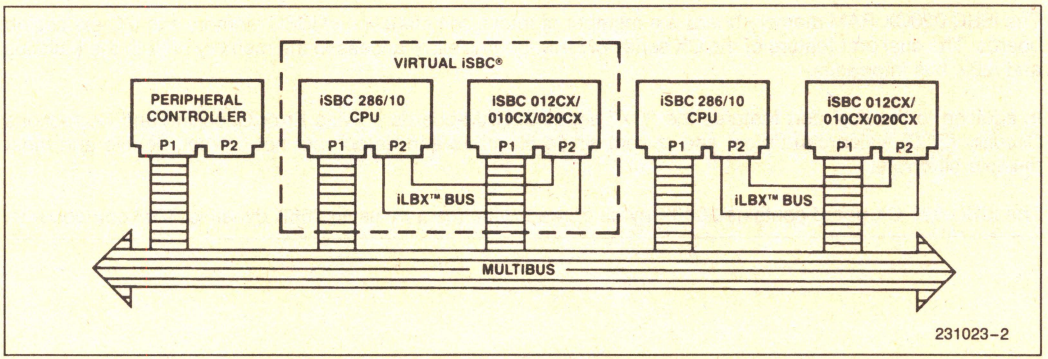


Figure 1. Typical iLBX System Configuration

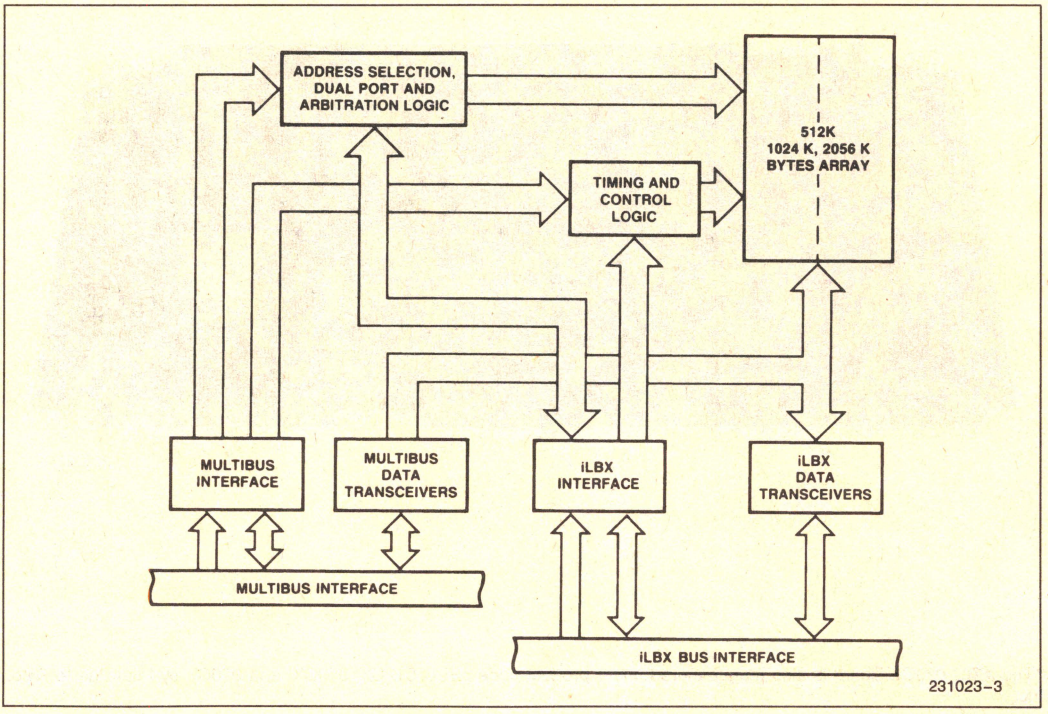


Figure 2. iSBC 020CX Block Diagram

System Memory Size

Maximum system memory size with this series of boards is 16 megabytes. Memory partitioning is independent for the Multibus interface and the iLBX interface.

For Multibus operations, on-board jumpers assign the board to one of four 4-megabyte pages. Each page is partitioned into 256 blocks of 16K bytes each. The smallest partition on any board in this series is 8K bytes. Jumpers assign the base address (lowest 16K block) within the selected 4-megabyte page.

The iLBX bus memory partitioning differs from the Multibus bus partitioning in that the iLBX bus address space consists of 256 contiguous blocks of 64K bytes totaling 16 megabytes. As with the Multibus bus partitioning, the base addresses are set with on-board jumpers.

Error Checking and Correcting (ECC)

Error checking and correction is accomplished with the Intel 8206 Error Checking and Correcting device. This ECC component, in conjunction with the ECC check bit RAM array, provides error detection and correction of single bit errors and detection only of double bit and most multiple bit errors. The ECC circuitry can be programmed via the Control Status Register (CSR) to various modes while error logging is supported by the Error Status Register (ESR). Both CSR and ESR communicate with the master CPU board through a single I/O port.

Battery Back-Up/Memory Protect

An auxiliary power bus is provided to allow separate power to the RAM array for systems requiring back-up of read/write memory. An active low TTL compatible memory protect signal is brought out on the auxiliary bus connector which, when asserted, disables read/write access to the RAM board. This input is provided for the protection of RAM contents during system power-down sequences.

SPECIFICATIONS

Word Size Supported

8- or 16-bits

Memory Size

2,097,152 bytes (iSBC 020CX board)

Access Times (All densities)

Multibus System Bus

Read/Full Write— 380 ns (max)
Write Byte — 530 ns (max)

iLBX Local Bus

Read/Full Write— 340 ns (max)
Write Byte — 440 ns (max)

Cycle Times (All densities)

Multibus System Bus

Read/Full Write— 490 ns (max)
Write Byte — 885 ns (max)

iLBX Local Bus

Read/Full Write— 375 ns
Write Byte — 740 ns

NOTE:
If an error is detected, read access time and cycle times are extended to 255 ns (max)

Memory Partitioning

Maximum System memory size is 16M Bytes for both Multibus and iLBX BUS. Multibus partitioning is by Page, Block and Base, while the iLBX BUS is by Block and Base only.

Page Address

Multibus— 0–4 megabytes; 4–8 megabytes, 8–12 megabytes; 12–16 megabytes

iLBX BUS — N/A

Base Address

Multibus System Bus— Any 16K byte boundary within the 4M-byte page.

iLBX Local Bus — Any 64K byte boundary selectable on board boundaries to 8M-bytes and some 64K-byte boundaries in the first megabyte. Others available if PAL programming is changed.

Power Requirements

Voltage—5 VDC ±5%

Product	Current	Standby (Battery Back-Up)
iSBC 020CX Board	5.3A (typ.) 7.5A (max.)	2.2A (typ.) 2.4A (max.)

Environmental Requirements

Operating Temperature: 0°C to 55°C airflow of 200 linear feet per minute

Operating Humidity: To 90% without condensation

Physical Dimensions

Width: 30.48 cm (12 inches)

Height: 17.15 cm (6.75 inches)

Thickness: 1.27 cm (0.50 inches)

Weight: 6589 gm (23.5 ounces)

Reference Manuals

144456-001 — Intel iLBX 020CX Specification

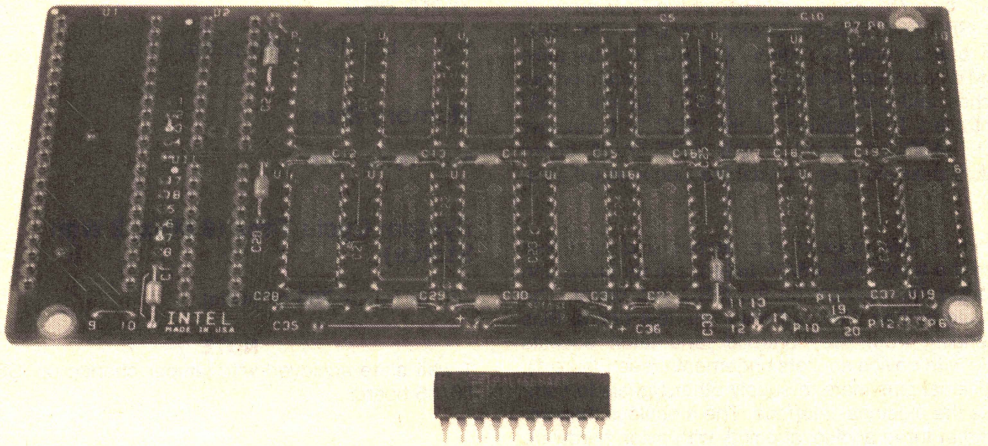
ORDERING INFORMATION

Part Number	Description
iSBC 020CX	2M byte RAM board with ECC

iSBC 314*
512K BYTE RAM MULTIMODULE BOARD

- **On-Board Memory Expansion for the iSBC 86/35 Single Board Computer**
- **iSBC 314 Module Provides 512K Bytes of Dual Port RAM Expansion for the iSBC 86/35 Board**
- **Reliable Mechanical and Electrical Interconnection**
- **Completes iSBC 86/35 Memory Array Providing a Full Megabyte Page of System Memory**
- **Increases System Throughput by Reducing Accesses to MULTIBUS Global Memory**
- **Low Power Requirements**
- **Battery Backup Capability**

The iSBC 314 512K byte RAM MULTIMODULE board provides simple, low cost expansion to double the on-board RAM capacity of the iSBC 86/35 Single Board Computer host to one megabyte. This RAM MULTIMODULE option offers system designers a simple, practical solution to expanding and improving the memory capability and performance of the iSBC 86/35 board. The iSBC 314 memory is configured on-board and can be accessed as quickly as the standard iSBC 86/35 memory, eliminating the need for accessing additional memory via the MULTIBUS system bus.



280000-1

9

*The iSBC 314 Board is also manufactured under product code pSBC 314 by Intel Puerto Rico, Inc.

September 1993
Order Number: 280000-003

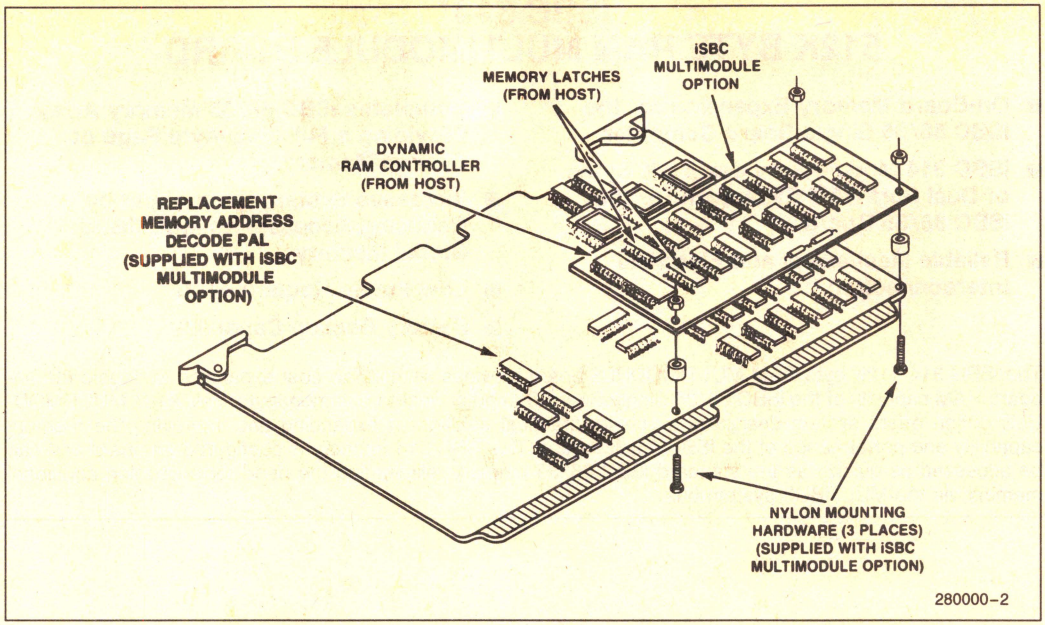


Figure 1. Installation of the MULTIMODULE RAM Module on the Host Single Board Computer

FUNCTIONAL DESCRIPTION

The iSBC 314 MULTIMODULE board measures 2.40″ by 5.75″ and mounts above the RAM array on the iSBC 86/35 Single Board Computer. The iSBC 314 board contains sixteen 256 Kbit x 1 dynamic RAM devices and three sockets; two for the memory latches and one for the Intel 8203 dynamic RAM controller. The addition of the iSBC 314 memory MULTIMODULE board to the iSBC 86/35 board makes possible a one megabyte single board solution.

To install the module, the latches and controller from the host iSBC 86/35 board, are removed and inserted into sockets on the iSBC 314 board. The module is then mounted onto the host board. Pins extending from the controller and latch sockets mate with device sockets underneath (see Figure 1). Additional pins mate to supply other signals to complete the electrical interface. The module is then secured at three additional points with nylon hardware to ensure the mechanical security of the assembly.

To complete the installation, one socketed PAL is replaced on the iSBC 86/35 board with the one supplied with the MULTIMODULE kit. This is the PAL which allows the host board logic to recognize its expanded on-board memory compliment.

SPECIFICATIONS

Word Size

8 or 16 bits (16-bit data paths)

Memory Size

512K bytes RAM

System Cycle Time (8 MHz, 2 Wait States)

750 ns (read); 750 ns (write)

NOTE:
1 wait state achieved with jumper change on iSBC 86/35 board.

Memory Addressing

iSBC 314 module with iSBC 86/35 board — 1M byte (total capacity); 0–FFFFFH. (See Figure 2, Memory Allocation)

Interface

The interface for the iSBC 314 MULTIMODULE board option is designed only for the iSBC 86/35 host board.

Wait-State Performance

A significant performance advantage of 2 wait-states is achieved when accessing memory on-board the iSBC 86/35 versus the performance of 6 wait-states when accessing memory off-board over the MULTIBUS. The iSBC 314 puts an additional 512K bytes of system memory on-board the iSBC 86/35 reducing the execution time by as much as 70%.

Memory Allocation

Segments of the combined host/MULTIMODULE RAM may be configured to be accessed either from off-board or on-board resources. The amount of memory allocated as either public or private resource may be configured in a variety of sizes. The address range boundaries for the 1 megabyte of RAM array of the iSBC 314 and iSBC 86/35 board combination are shown in Figure 2 for accesses from both on-board and off-board resources.

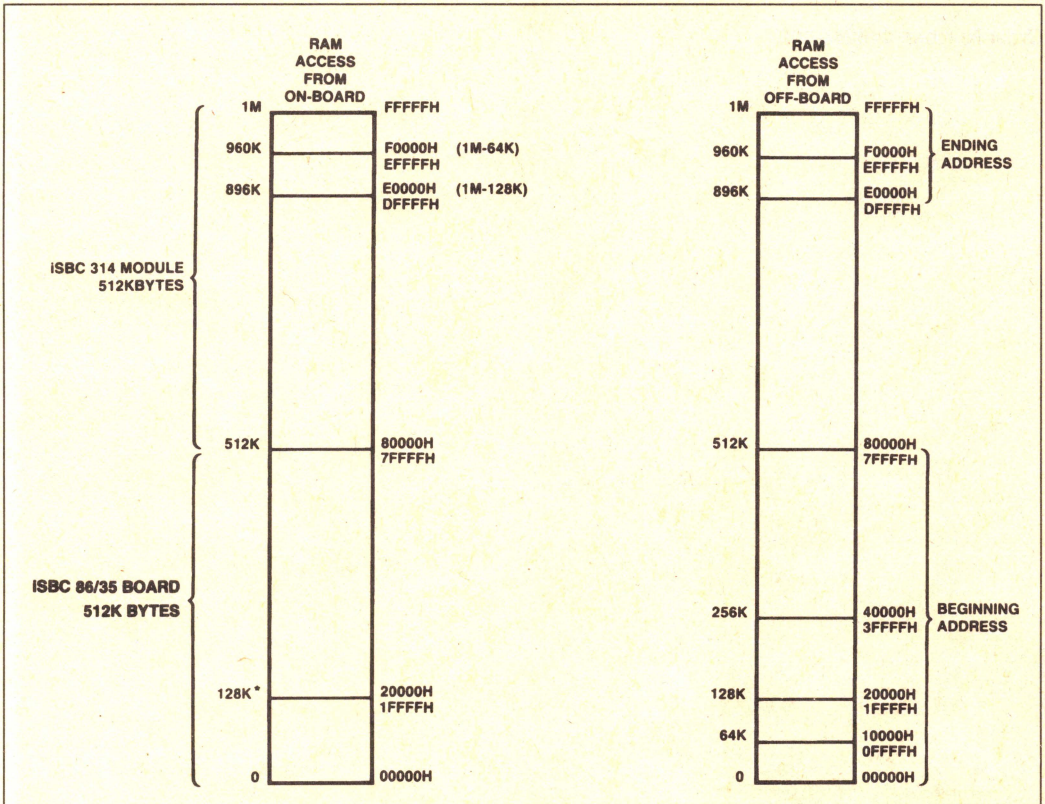
Auxiliary Power

The low power memory protection option included on the iSBC 86/35 board supports the iSBC 314 module.

Physical Characteristics

Width: 2.4 in. (6.10 cm)
Length: 5.75 in. (14.61 cm)
Depth*: 0.72 in. (1.83 cm)
Weight: 0.13 oz. (59g)

NOTE:
*Combined depth including host board.



NOTE:
All memory above this boundary may be disabled under software control to allow access to MULTIBUS system bus.

Figure 2. Address Range Selection

9

Electrical Characteristics

DC Power Requirements*

*Additional power required by the iSBC 314 MULTIMODULE is:

Typical: 60 mA @ +5V

Maximum: 140 mA @ +5V

Environmental Characteristics

Operating Temperature: 0°C to +55°C

Relative Humidity: to 90% (without condensation)

Reference Manual

All necessary documentation for the iSBC 314 MULTIMODULE board is included in the iSBC 86/35 Hardware Reference Manual (NOT SUPPLIED).

Order Number: 146245-002

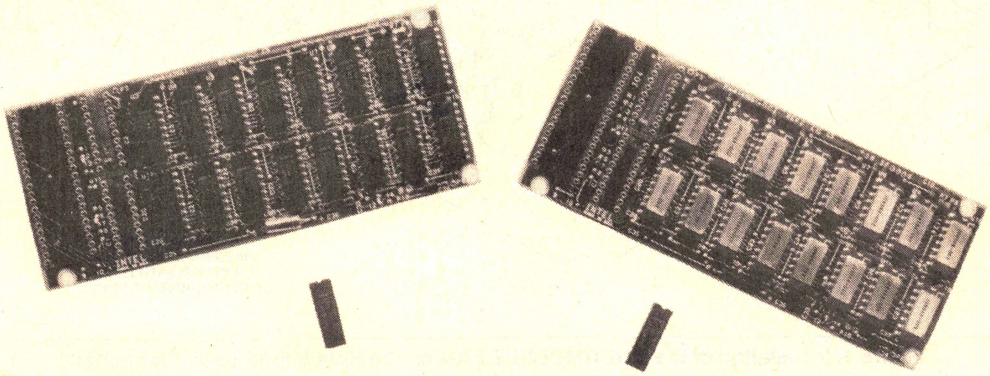
ORDERING INFORMATION

Part Number	Description
SBC 314	512K byte Memory MULTIMODULE option for iSBC 86/35 board

iSBC 304*
128K BYTE RAM MULTIMODULE BOARD

- **iSBC 304 Module Provides 128K Bytes of Dual Port RAM Expansion for the iSBC 86/30 or iSBC 86/35 Board**
- **On-board Memory Expansion Eliminates MULTIBUS System Bus Latency and Increases System Throughput**

The iSBC 304 module provides simple, low cost memory expansion for the iSBC 86/30 and iSBC 86/35 Single Board Computers. The iSBC 304 provides 128K bytes RAM expansion to the iSBC 86/35 giving a total capacity of 640K bytes RAM memory. The RAM MULTIMODULE option offers flexibility in defining and implementing Intel single board computer systems. RAM MULTIMODULES expand the memory configuration on-board, eliminating the need for accessing the additional memory via the MULTIBUS system bus.



210329-1

9

*The iSBC 304 Board is also manufactured under product code pSBC 304 by Intel Puerto Rico, Inc.

September 1993
Order Number: 210329-003

FUNCTIONAL DESCRIPTION

Each MULTIMODULE contains dynamic RAM devices and sockets for the dynamic RAM controller. To install the module, the latches and controller from the host CPU board are removed and inserted into sockets on the RAM MULTIMODULE. The module is then mounted onto the host board. Pins extending from the controller and latch sockets mate with device sockets underneath (see Figure 1). Additional pins mate to supply other signals to complete the electrical interface.

The module is then secured at three additional points with nylon hardware to ensure the mechanical security of the assembly.

To complete the installation, one socketed PROM is replaced on the host CPU board with the one supplied with the MULTIMODULE kit. This is the MULTIBUS address decode PROM which allows the host board logic to recognize its expanded on-board memory compliment.

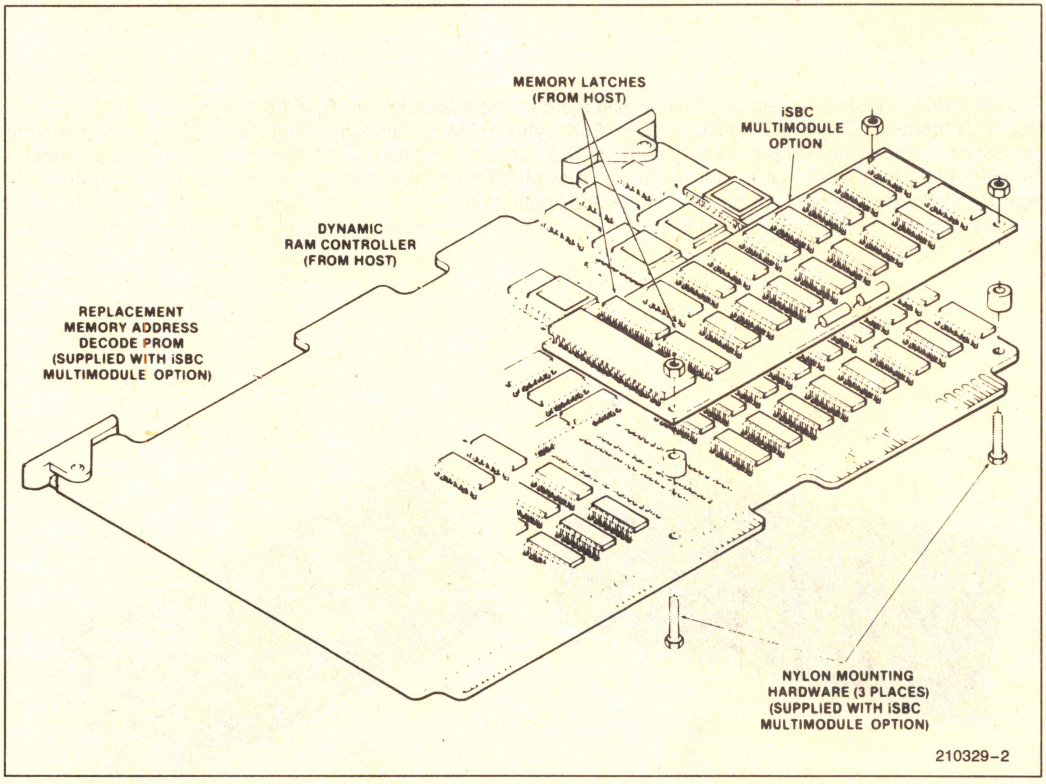


Figure 1. Installation of the MULTIMODULE RAM on the Host Single Board Computer

SPECIFICATIONS

Word Size

8 or 16 bits (16-bit data paths)

Memory Size

iSBC 304 Module—128K bytes RAM

Cycle Time

iSBC 304—700 ns (read); 700 ns (write)

Memory Addressing

CPU ACCESS

iSBC 304 (with iSBC 86/35)—640K bytes (total capacity); 0–9FFFFH (address range)

iSBC 304 (with iSBC 86/30)—256K bytes (total capacity); 0–3FFFFH (address range)

MULTIBUS Access

Jumper selectable for any 32K byte boundary, but not crossing a 256K byte boundary.

Private Memory Allocation

Segments of the combined host/MULTIMODULE RAM memory may be configured as a private resource, protected from MULTIBUS system access. The amount of memory allocated as a private resource may be configured in increments of 25% of the total on-board memory ranging from 0% to 100%.

Auxiliary Power

The low power memory protection option included on the CPU host boards supports the RAM modules.

Physical Characteristics

Width: 2.4 in. (6.10 cm)

Height: 5.75 in. (14.61 cm)

Depth*: 0.72 in. (1.83 cm)

Weight: 0.13 oz. (59 g)

***NOTE:**
Combined depth including host board.

Electrical Characteristics

DC POWER REQUIREMENTS

iSBC 304: 640 mA at +15V incremental power

Environmental Characteristics

Operating Temperature: 0°C to 55°C

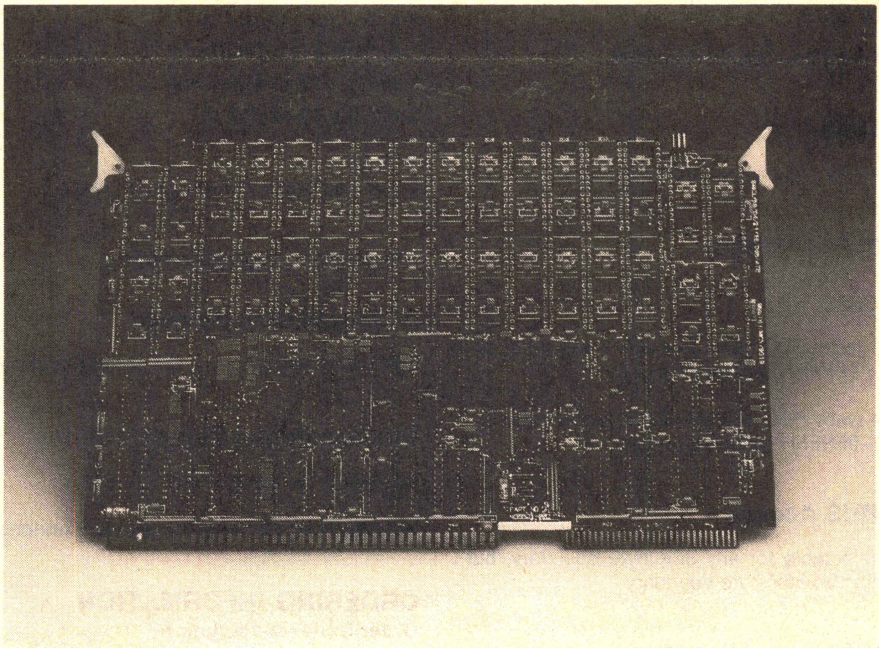
Relative Humidity: to 90% (without condensation)

ORDERING INFORMATION

Order Code	Description
SBC304	128K MULTIMODULE option for iSBC 86/30 or iSBC 86/35 CPU boards

9

iSBC 429 UNIVERSAL SITE MEMORY EXPANSION BOARD



280668-1

CMOS MULTIBUS I MEMORY EXPANSION BOARD SUPPORTS LATEST MEMORY TECHNOLOGY

The iSBC 429 board provides a wide range of memory expansion capabilities for Multibus designs. Up to 4 MBytes of memory can be installed using EPROM, Flash memory, SRAM, E^{2}PROM or Static NVRAM.

The CMOS implementation of the iSBC 429 makes it ideal for low power applications.

All of Intel's Single Board Computers can communicate with the iSBC 429 using the Multibus System bus. Alternatively, the iSBC 429 may be optionally configured to use the iLBX bus for faster access to the iSBC 186/03A, 286/10A, 286/12 series or 386/12 series of Single Board Computers.

FEATURES

- Supports EPROM, Page Mode EPROM, E^{2}PROM, Flash Memory, SRAM and Static NVRAM
- Thirty-two standard 32-pin JEDEC sites (supports both 28-pin and 32-pin devices) up to 4 MByte capacity
- iLBX Bus or Multibus Configurability
- Low power CMOS design
- Battery Backup/Memory Protect support
- Assignable anywhere within a 16 Megabyte address space on 4 Kbyte boundaries

August 1993
Order Number: 280668-001

FEATURES

iLBX BUS

The iSBC 429 board can be configured via jumpers to communicate with either the Multibus interface or the iLBX Bus interface. Significant memory access time improvements can be realized using the iLBX Bus interface versus the Multibus interface, due to its dedicated, unarbitrated architecture. Additional information on the iLBX Bus is available in the iLBX Specification, order number 145695-Rev. A.

CMOS DESIGN

For embedded control applications which are sensitive to power consumption, the iSBC 429 was designed with CMOS components and it will support many CMOS memory devices. Unpopulated, the iSBC 429 requires 5.25 watts at 5 volts.

FLASH MEMORY SUPPORT

The iSBC 429 board supports Intel's new CMOS Flash Memory devices. These new memory devices offer the most cost-effective and reliable alternative for updatable non-volatile memory. Memory contents can be erased and reprogrammed on-board during subassembly test, in-system during final test, and in-system after sale.

MEMORY BANKS

The thirty-two sites on the iSBC 429 board are partitioned into two banks of 16 sites each. Both banks are independently configurable to any of the device types supported on the board. Each bank can support up to 2 Megabytes using 27010 devices.

MEMORY ADDRESSING

The address space of each bank can be independently configured for starting address and size. The starting address can be on any 4 KByte boundary within the 16 MByte Multibus address space. The size of each bank is a multiple of 64 KBytes.

MODE OF OPERATION

The iSBC 429 board can operate in one of two modes: the 8 bit only mode or the 8/16 bit mode. The 8 bit mode provides the most efficient memory configuration for systems handling 8 bit data only. The 8/16 bit mode allows the iSBC 429 board to be compatible with systems employing 8 bit and 16 bit masters. The mode of operation is selected by on-board jumpers and is available for both Multibus and iLBX Bus configurations.

MEMORY ACCESS

The iSBC 429 board has jumper-selectable access times for each bank which allows the board to be tailored to the performance of the particular devices which are installed in the iSBC 429 board. The iSBC 429 accepts devices with an access time ranging from 150 ns with a minimum granularity of 99 ns and results in a board access time from 182 ns to 1667 ns. Each bank can be configured for access time.

INHIBITS

Inhibit signals are provided on the iSBC 429 board to allow ROM to overlay RAM for bootstrapping or diagnostic operations. Each bank of the iSBC 429 board can be overlayed with the system RAM by jumpers provided on the board. (i.e. If banks are overlapped, inhibits can be used to select the appropriate bank.)

BATTERY BACKUP

The iSBC 429 board supports battery backup operation via a connector on the board. An auxiliary power bus is provided to allow separate power to the memory array for systems requiring battery backup. Selection of this auxiliary power bus is made via jumpers on the board.

An active-low TTL compatible Memory Protect signal is brought out on the auxiliary connector which, when asserted, disables access to the memory array. This input is provided for the protection of Memory contents during system power-down sequences.

SPECIFICATIONS

MEMORY DEVICES SUPPORTED BY THE iSBC 429								
Type	Size							
	8K×8	16K×8	32K×8	64K×8	128K×8	256K×8	4×16K×8	8×16×8
EPROM	2764	27128	27256	27512	27010	27020	—	—
ROM	✔[1]	✔	✔	✔	✔	✔	—	—
Page Mode EPROM	—	—	—	—	—	—	27513	27011
E²PROM[2]	2864A	—	—	—	—	—	—	—
Flash27F64 Memory[3]		—	27F256 28F256	—	—	—	—	—
Static NVRAM[4]	✔	—	✔	—	—	—	—	—
SRAM	✔	—	✔	—	—	—	—	—

1 "✔" denotes that the iSBC 429 board will support the device indicated, but that it is not currently available from Intel.
2 Five Volt only, Enhanced
3 12 Volt V_{pp} only
4 Static NVRAM devices exceed the height specification for Multibus. The iSBC 429 will occupy more than one slot with these devices installed.

WORD SIZE

8 or 8/16 bits

MEMORY SIZE

Sockets are provided for up to thirty-two 32-pin or 28-pin devices which can provide up to 4 Megabytes of EPROM/ROM/SRAM/Flash Memory.

ACCESS TIME

Access time is jumperable from 182 ns to 1667 ns with a granularity of 99 ns to optimize performance for the devices which are installed and is equivalent for Multibus and iLBX Bus.

POWER REQUIREMENTS

V_{cc} = 5 volts ±5%
V_{pp} = 12 volts ±5%
I_{cc} = 1.2 amps, maximum, without any memory devices in the board.

PHYSICAL CHARACTERISTICS

Width - 12.00 inches (30.48 cm)
Depth - 7.05 inches (17.91 cm)
Height - .5 inches (1.27 cm)

ENVIRONMENT

Operating Temperature - 0° C to + 60° C (Convection cooling)
Relative Humidity - 90% non-condensing

ORDERING INFORMATION

PART NUMBER	DESCRIPTION
SBC 429	Universal Site Memory Expansion Board

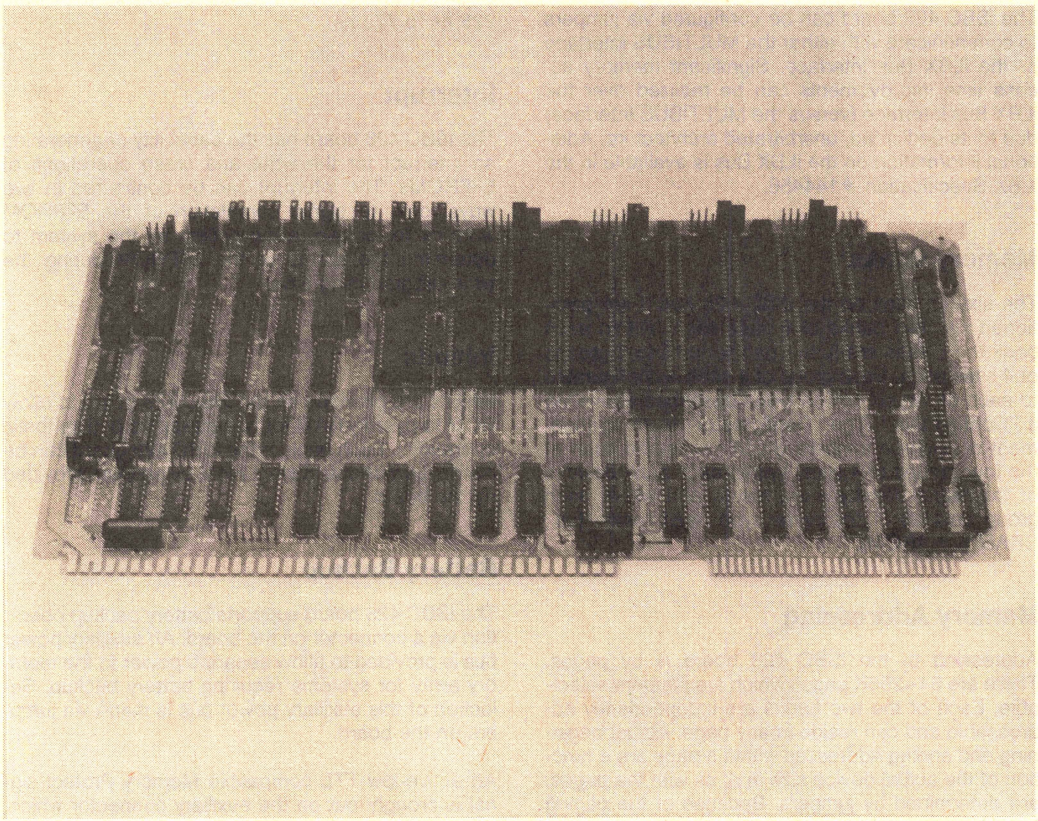
REFERENCE MANUAL

457317-001 - iSBC 429 Hardware Reference Manual (NOT SUPPLIED)

iSBC 428 UNIVERSAL SITE MEMORY EXPANSION BOARD

- **Supports EPROM, ROM, E²PROM, SRAM, IRAM and NVRAM**
- **iLBX BUS or MULTIBUS Selectable**
- **Provides Support for Battery Backup/ Memory Protect**
- **Sixteen 28-Pin Universal Sites**
- **Assignable Anywhere within a 16 Mbyte Address Space on 256K Byte Boundries**
- **Jumper Selectable Base Address on 4K Byte Boundaries**

The iSBC 428 Universal Site Board is a member of Intel's complete line of Memory and I/O Expansion boards. The iSBC 428 Universal Site Memory Expansion Board interfaces directly to the iSBC 80, iSBC 88, or iSBC 86 Single Board Computers via the MULTIBUS System Bus to expand system memory requirements, while system memory requirements for iSBC 286 Single Board Computer can interface via either the MULTIBUS or the high speed iLBX Bus.



281013-1

9

September 1993
Order Number: 281013-001

FUNCTIONAL DESCRIPTION

General

The iSBC 428 board contains sixteen 28 pin sockets. The actual capacity of the board is determined by the type and quantity of components installed by the user. The iSBC 428 board is compatible with five different types and densities of devices: the 2K by 8 thru 64K by 8 EPROM/ROM devices, 2K by 8 thru 8K by 8 "Five Volt Only, Enhanced" E^2PROM devices, 512 by 8 thru 16K by 8 NVRAM (Non-Volatile RAM) devices, 2K by 8 thru 32K by 8 SRAM devices, and 8K by 8 IRAM (Integrated RAM) devices. In addition the board can be accessed by either the MULTIBUS System Bus or Intel's new high speed iLBX Bus.

iLBX Bus

The iSBC 428 board can be configured via jumpers to communicate with either the MULTIBUS interface or the iLBX Bus interface. Significant memory access time improvements can be realized over the iLBX Bus interface (versus the MULTIBUS interface) due to its dedicated, unarbitrated architecture. Additional information on the iLBX Bus is available in the iLBX Specification #144456.

Memory Banks

The sixteen sites on the iSBC 428 board are partitioned into two banks of 8 sites each. Within each bank the 8 sites are futher partitioned into 2 groups of 4 sites each. Each group of 4 sites is configurable to each of the six device types described above via a "Configurator". The "Configurator" is an arrangement of push-on jumpers which configures each of the four groups of 4 sites. Within each bank devices of the same density must reside and within each group devices of the same type must reside (i.e., SRAM or EPROM).

Memory Addressing

Addressing of the iSBC 428 board is by pages. There are 64–256K pages which are jumpers selectable. Each of the two banks are independently addressable and can reside in any page. Actual beginning and ending addresses within a page are a function of the actual device size and, as with the pages, are determined by jumpers. Because of the paging based memory addressing architecture more than one iSBC 428 board can be placed in a system.

Mode of Operation

The iSBC 428 board can operate in one of two modes: the 8 bit only mode or the 8/16 bit mode. The 8 bit mode provides the most efficient memory configuration for systems handling 8 bit data only. The 8/16 bit mode allows the iSBC 428 board to be compatible with systems employing 8 bit and 16 bit masters. The mode of operation is selected by on board jumpers and is available for both MULTIBUS and iLBX Bus configurations.

Memory Access

The iSBC 428 board has jumper selectable access time, which allows the board to be tailored to the performance of the particular devices which are installed in the iSBC 428 board. The board can be configured via jumpers to accept devices with an access time range of 50 ns to 500 ns with a granularity of 50 ns and results in a board access time from 225 ns to 775 ns.

Interrupt

The iSBC 428 board has the capability of generating an interrupt for the write and erase operations of E^2PROMs. The interrupt can be configured in two ways: one, to signal completion of the E^2PROM write cycle, or two, allow polling by the system to determine the status of the E^2PROM during the write programming time.

Inhibits

Inhibits are provided on the iSBC 428 board to allow ROM to overlay RAM for bootstrapping or diagnostic operations. Each bank of the iSBC 428 board can be overlayed with the system RAM by jumpers provided on the board.

Battery Backup

The iSBC 428 board supports battery backup operation via a connector on the board. An auxiliary power bus is provided to allow separate power to the memory array for systems requiring battery backup. Selection of this auxiliary power bus is made via jumpers on the board.

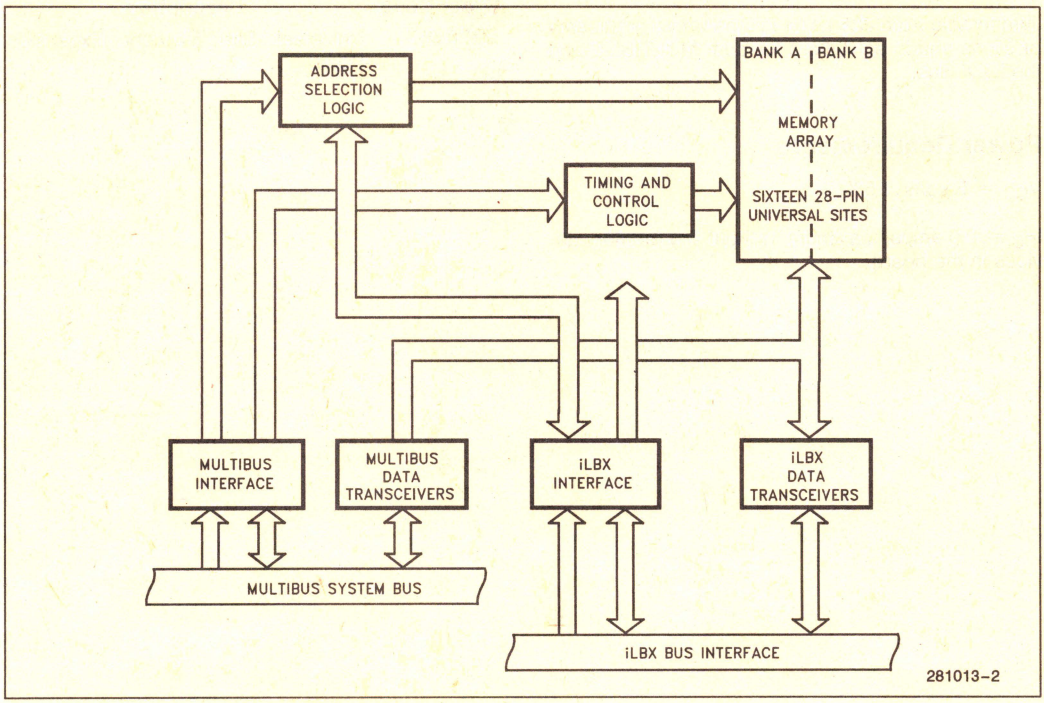
An active-low TTL compatible Memory Protect signal is brought out on the auxiliary connector which, when asserted, disables access to the memory array. This input is provided for the protection of Memory contents during system power-down sequences.

Devices Supported

Listed below are the current and future devices supported by the iSBC 428 board.

	Size							
Type	**512 x 8**	**2K x 8**	**4K x 8**	**8K x 8**	**16K x 8**	**32K x 8**	**64K x 8**	**Comments**
EPROM	—	2716	2732A	2764	27128	27256	27512	—
ROM	—	X	X	X	X	X	X	—
EEPROM	—	2817A	X	X	X	X	—	5V, Enhanced
SRAM	—	X	X	X	X	X	—	NMOS & CMOS
NVRAM	—	X	X	X	—	—	—	—
IRAM	—	—	—	2186	—	X	—	—

X—Denotes that the iSBC 428 board will support the device indicated but that it is not currently available from Intel.



iSBC 428 Block Diagram

SPECIFICATIONS

Word Size

8 or 8/16 bits

Memory Size

Sockets are provided for up to sixteen 28 pin devices which can provide up to 512K bytes of EPROM/ROM/SRAM.

Access Time

Jumperable from 225 ns to 775 ns with a granularity of 50 ns and is equivalent for both MULTIBUS and the iLBX Bus.

Power Requirements

V_{CC} = 5 volts ±5%

I_{CC} = 2.0 amps, maximum, without any memory devices in the board.

Physical Characteristics

Length: 30.48 cm (12 inches)
Width: 17.15 cm (7.05 inches)
Depth: 1.27 cm (0.5 inches)

Environment

Operating Temperature: 0°C to +55°C
Relative Humidity: 90% non-condensing

ORDERING INFORMATION

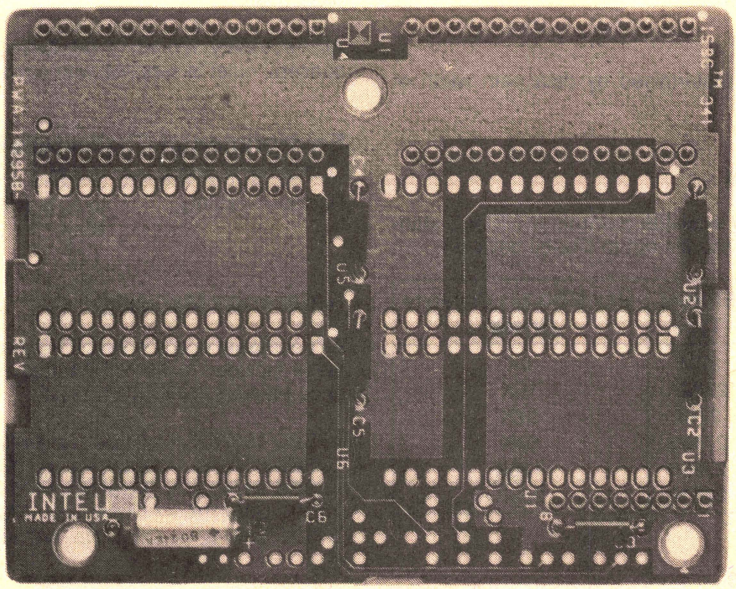
Order Code	Description
SBC428	Universal Site Memory Expansion Board

iSBC 341
28-PIN MULTIMODULE EPROM

- **On-board Memory Expansion for iSBC 86/05A, iSBC 88/45, iSBC 186/03A, iSBC 286/10A, iSBC 188/56, iSBC 286/12 Series, and iSBC 88/40A Microcomputers**
- **Supports JEDEC 24/28-Pin Standard Memory Devices, Including EPROMs, Byte-Wide RAMs, and E²PROMs**
- **Sockets for Up to 256K Bytes of Expansion with Intel 27512 EPROMs**
- **On-Board Expansion Provides "No Wait State" Memory Access with Selected Devices**
- **Simple, Reliable Mechanical and Electrical Interface**

The iSBC 341 28-pin MULTIMODULE EPROM board provides simple, low-cost expansion of the on-board EPROM capacity of the iSBC 86/05A, the iSBC 88/25, iSBC 186/03A, iSBC 286/10A, iSBC 286/12 Series Single Board Computers and the iSBC 88/40A Measurement and Control Computer. Four additional 28-pin sockets support JEDEC 24/28-pin standard devices, including EPROMs, byte-wide static and psuedo-static RAMs.

The MULTIMODULE expansion concept provides the optimum mechanism for incremental memory expansion. Mounting directly on the microcomputer, the benefits include low cost, no additional power requirements beyond the memory devices, and higher performance than MULTIBUS-based memory expansion.



280214-1

9

FUNCTIONAL DESCRIPTION

The iSBC 341 28-pin MULTIMODULE EPROM option effectively doubles the number of sockets available for EPROM on the base microcomputer board on which it is mounted. The iSBC 341 board contains six 28-pin sockets. Two of the sockets have extended pins which mate with two of the sockets on the base board. Two of the EPROMs which would have been inserted in the base board are then reinserted in the iSBC 341 sockets. Additional interface pins also connect chip select lines and power. The mechanical integrity of the assembly is assured with nylon hardware securing the unit in two places.

Through its unique interface, the iSBC 341 board can support 8- or 16-bit data paths. The data path width is determined by the base board—being 8 bits for the iSBC 88/40A and iSBC 88/25 microcomputers, and 8/16 bits for the iSBC 86/05A, iSBC 186/03A, iSBC 286/10A, and iSBC 286/12 Series Single Board Computers.

SPECIFICATIONS

Word Size

8 or 8/16 bits (determined by data path width of base board).

Memory Size

256K bytes with available technology (JEDEC standard defines device pin-out to 512-bit devices).

Device Size (Bytes)	EPROM Type	Max iSBC 341 Capacity (Bytes)
2K x 8	2716	8K
4K x 8	2732A	16K
8K x 8	2764	32K
16K x 8	27128	64K
32K x 8	27256	128K
64K x 8	27512	256K

Access Time

Varies according to base board and memory device access time. Consult data sheet of base board for details.

Memory Addressing

Consult data sheet of base board for addressing data.

POWER REQUIREMENTS

Devices(1)	Max Current @ 5V ±5%
2716	420 mA
2732A	600 mA
2764	600 mA

NOTE:
1. Incremental power drawn from host board for four additional devices.

Auxiliary Power

There are no provisions for auxiliary power (battery backup) on the iSBC 341 option.

Physical Characteristics

Width: 3.4 in. (8.64 cm)

Length: 2.7 in. (6.86 cm)

Height: 0.78 in. (1.98 cm)*

Weight: 5 oz. (141.5 gm)

*Includes height of mounted memory devices and base board.

All necessary mounting hardware (nylon screws, spacers, nuts) is supplied with each kit.

Environmental Characteristics

Operating Temperature: 0°C to +55°C

Relative Humidity: to 90% (without condensation)

Reference Manuals

All necessary documentation for the iSBC 341 module is included in the CPU board Hardware Reference Manuals (NOT SUPPLIED)

iSBC 186/03A — Order No. 148060-001
iSBC 86/05A — Order No. 147162-002
iSBC 88/40A — Order No. 147049-001
iSBC 286/10A — Order No. 147532-001
iSBC 286/12 — Order No. 147533-001

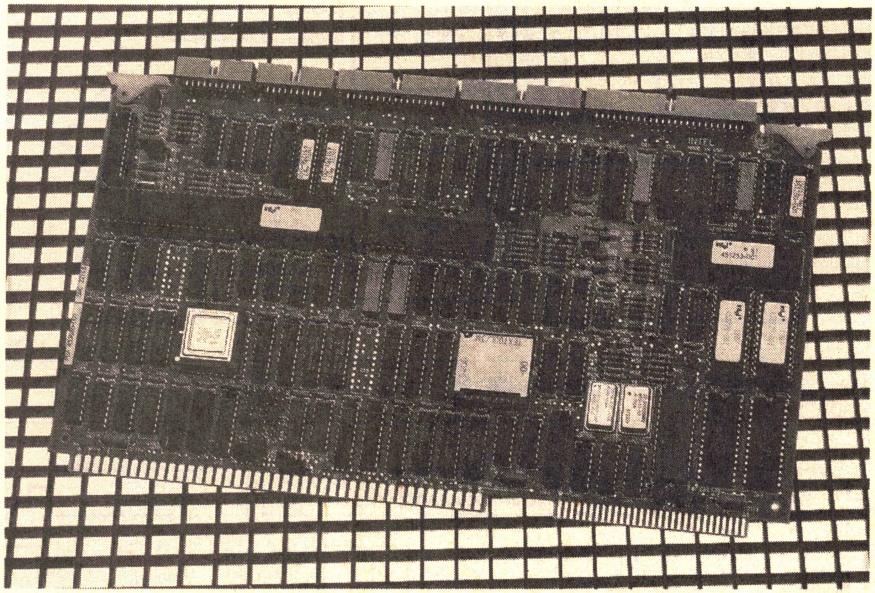
ORDERING INFORMATION

Part Number	Description
SBC 341	28-Pin MULTIMODULE EPROM

Multibus I Peripheral Controllers

10

iSBC 221* PERIPHERAL CONTROLLER



280410-1

MULTIBUS I CONTROLLER FOR HIGH PERFORMANCE, HIGH CAPACITY PERIPHERALS

The iSBC 221 is a multifunction peripheral controller that provides access to high-performance, high-capacity disk drives (hard, flexible, and streaming tape). I/O bound applications and/or those requiring high disk capacity will especially benefit from this fast, reliable controller. The iSBC 221 can replace the Intel iSBC 214 without changing the operating system device driver or the disk drives.

FEATURES

- Support for ESDI and ST506/412 hard disk drives, SA 45X/46X/475 flexible disk drives, and QIC-02 streaming tape drives
- Multiple track caching via 128K on-board data buffer
- Dual bus structure
- 10 MHz 80186 Microprocessor
- Mirror backup/restore between tape and hard drive
- On-board self-test diagnostics
- Error-checking and correcting code logic
- Support for 4,096 cylinders and 16 heads

*The iSBC 221 is also manufactured under product code pSBC 221 by Intel Puerto Rico, Inc.
*XENIX is a trademark of Microsoft, Inc.
UNIX is a trademark of UNIX System Laboratories.

August 1993
Order Number: 280410-002

FEATURES

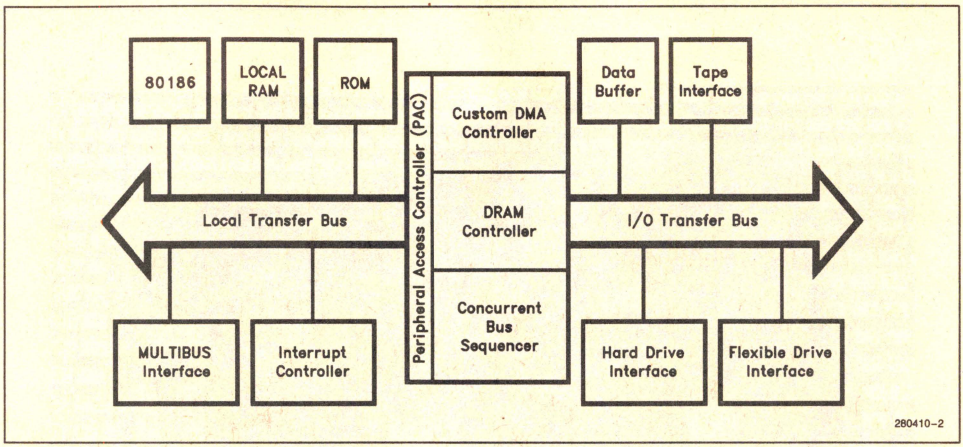


Figure 1: Simplified Block Diagram of iSBC 221 Peripheral Controller

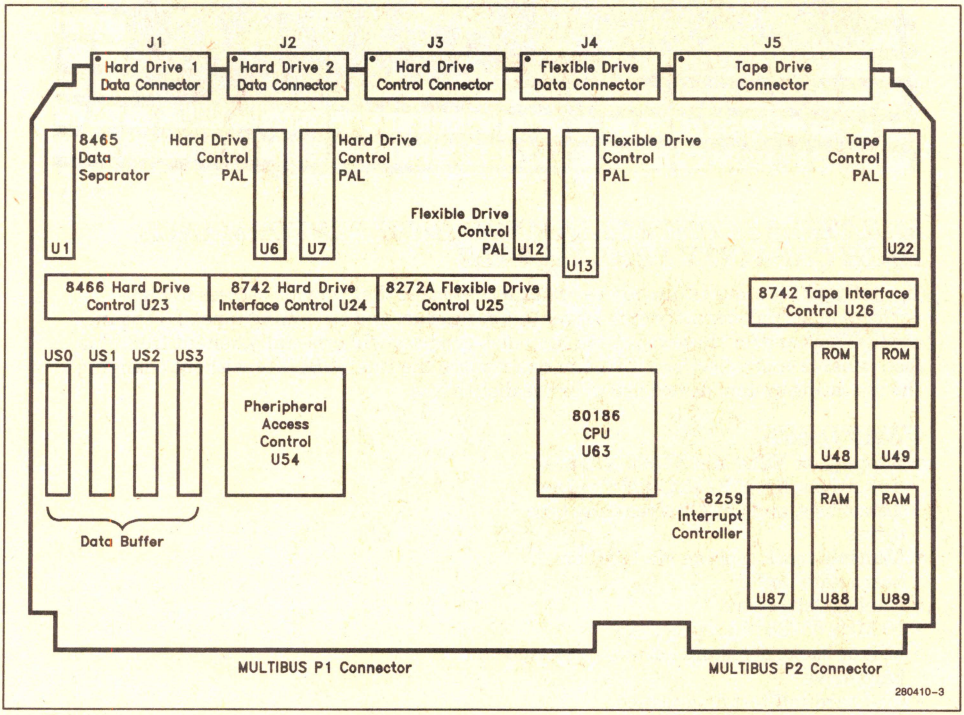


Figure 2: Connectors and Major Components of iSBC 221 Peripheral Controller

FEATURES

INTERFACE SUPPORT

	Interface	Transfer Rate
Hard Disk (up to 2)	ESDI	up to 15 Mbit/sec
	ST506/412	5 Mbit/sec
Flexible Disk (up to 4)	SA 475	250/500 Kbit/sec
	SA 460/465	125/250 Kbit/sec
	SA 450/455	125/250 Kbit/sec
Streaming Tape (up to 4)	QIC-02	90/112.5 Kbit/sec (typical)

HIGH PERFORMANCE

I/O-bound applications are accelerated by the combination of the ESDI standard, a 128K data buffer, a 10 MHz 80 186 microprocessor, and a dual bus structure. The dual bus structure allows the iSBC 221 to concurrently transfer data between the controller and the peripheral devices and between the controller and the host.

WORLDWIDE SERVICE AND SUPPORT

Intel provides support for board repair or on-site service. Development support options include phone support, subscription service, on-site consulting and customer training.

QUALITY AND RELIABILITY TESTING

The iSBC 221 is designed and manufactured in accordance with Intel's high quality standards. We then verify quality through rigorous testing in our state-of-the-art Environmental Test Laboratory.

SPECIFICATIONS

PHYSICAL CHARACTERISTICS

Length: 12.0 in. (304.8 mm)
Width: 6.75 in. (171.5 mm)
Approximate Weight: 24 oz. (680 g)

POWER REQUIREMENTS

+5 VDC @ 4.5A maximum
±12V @ 0.5A

ENVIRONMENTAL REQUIREMENTS

Operating Temperature: 0°C to +55°C @ 200 LFM
Non-Operating: −55°C to +85°C
Humidity: 0% to 90% non-condensing

REFERENCE MANUAL

iSBC 221 Peripheral Controller User's Guide
Order Number 451210

DEVICE DRIVERS

Check the latest release of the following operating systems for details:

iRMX I
iRMX II
iRMX III
XENIX*
UNIX*

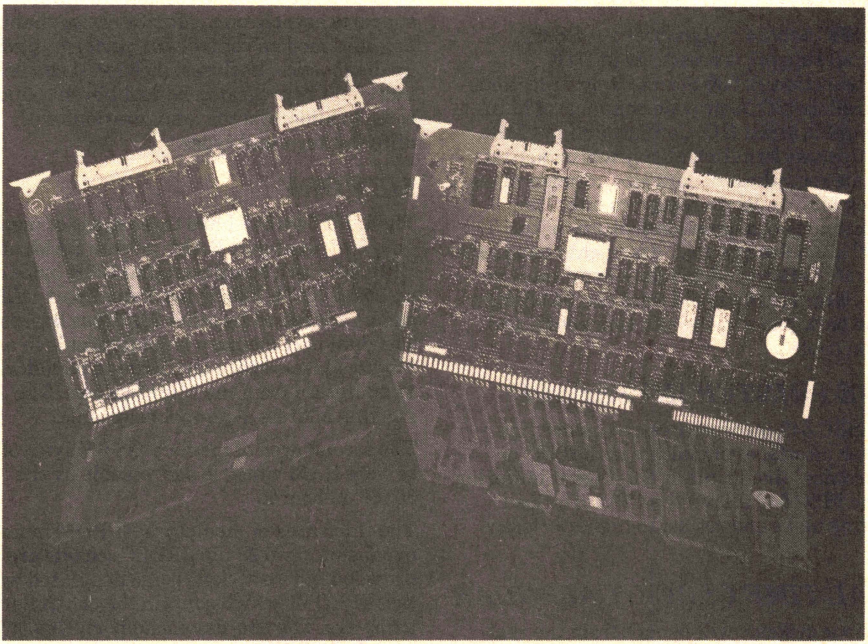
ORDERING INFORMATION

Order Code	Description
SBC221	Peripheral Controller

Multibus I Serial Communication Boards

11

iSBC 548/549 TERMINAL CONTROLLERS



280674-1

HIGH PERFORMANCE TERMINAL CONTROLLER BOARDS FOR MULTIBUS I

The iSBC 548 and iSBC 549 are intelligent terminal controllers for Multibus I applications. The iSBC 548 provides basic multiuser support with 8 channels of RS 232 Asynchronous interface. The iSBC 549 combines 4 serial channels with a real-time clock and a line printer interface. Acting as intelligent slaves for communication expansion, these boards provide high performance, low cost solutions for multi-user systems.

FEATURES

iSBC 548 Features

- Supports eight channels asynchronous RS232 interface

iSBC 549 Features

- Supports four channels asynchronous RS232 interface
- Line printer interface
- Real-time clock/calendar with battery backup

Standard iSBC 548/549 Features

- 8 MHz 80186 Microprocessor
- Supports transfer rates up to 19.2K Baud
- 128 Kbytes Zero Wait State DRAM (32K Dual Port)
- Supports Full Duplex Asynchronous Transmissions
- Jumper selectable memory mapping, I/O mapping and Multibus Interrupts

*IBM is a trademark of International Business Machines.
*Centronics is a registered trademark of Centronics, Inc.

August 1993
Order Number: 280674-002

FEATURES

ASYNCHRONOUS RS232 INTERFACE SUPPORT

The iSBC 548/549 Asynchronous RS232 Internal support is presented in DTE Configuration. 82530 Serial Communications Controllers (SCCs) provide channels of half/full duplex serial I/O. Configurability of the 82530 allows handling all asynchronous data formats regardless of data size, number of start or stop bits, or parity requirements. The synchronous transmission features of the 82530 are not supported. An on-chip baud rate generator allows independent baud rates on each channel. The serial lines can be brought to the back-panel via 40-pin connectors and ribbon cable.

LINE PRINTER INTERFACE

The iSBC 549 incorporates a standard line printer interface compatible with IBM* or Centronics* line printers. Intelligent buffering on the iSBC 549 allows the CPU to offload printing tasks and return to higher priority jobs.

REAL-TIME CLOCK/CALENDAR

Multibus systems will benefit from the real-time clock present on the iSBC 549 in applications requiring time stamp operations, unattended boots and other calendar requirements. The clock/calendar circuit is backed up by a non-rechargeable battery which keeps the clock/calendar operating for six months with all other power off.

8 MHZ 80186 MICROPROCESSOR

The 80186 central processor component provides high- performance, flexibility, and powerful processing. The 80186/82530 combination with on-board PROM/EPROM sites, and dual-port RAM provides the intelligence and speed to manage multi-user communications.

TRANSFER RATES UP TO 19.2K BAUD

Collectively, each board has dual-port RAM providing an on-board buffer to handle incoming and outgoing messages at data rates up to 19.2K baud. The resident firmware supports asynchronous RS232 serial channels, provides modem control and performs power-up diagnostics. Each serial channel can be individually programmed to different baud rates to allow system configurations with differing terminal types.

MEMORY

The iSBC 548/549 have three areas of memory on-board: dual-port RAM, private RAM, and EPROM. Each board contains 128K bytes of on-board RAM, 32K bytes of dual-port RAM can be addressed by other Multibus boards. The dual port memory is configurable in a 16 Mbyte address space on 32 Kbyte boundaries as addressed from the Multibus port. The starting address is jumper selectable.

The second area of memory is 96 Kbytes of private RAM which is addressable by the 80186 on-board.

The third area of memory is EPROM memory expansion. Two 28-pin JEDEC sockets are provided. These sockets come populated with two EPROMs which contain the controller firmware. The boards can support 2764, 27128 and 27256 EPROMs, giving a total capacity of 64 Kbytes. The EPROM runs with zero wait states if EPROMs of access times 250ns or less are used. No jumper changes are needed to access different size EPROMs.

WORLDWIDE SERVICE AND SUPPORT

Intel provides support for board repair or on-site service. Development options include phone support, subscription service, on-site consulting, and customer training.

QUALITY AND RELIABILITY

The iSBC 548 and iSBC 549 are designed and manufactured in accordance with Intel's high quality standards. We then verify quality through rigorous testing in our state-of-the-art Environmental Test Laboratory.

FEATURES

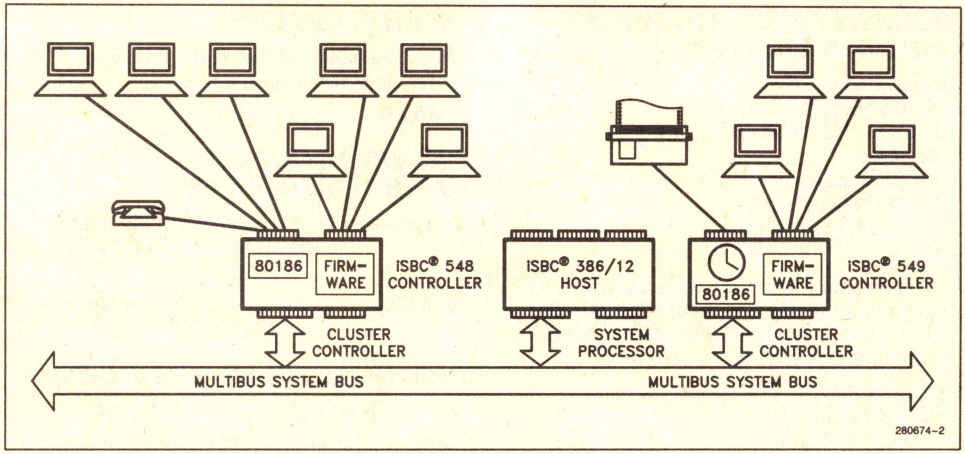


Figure 1: Terminal/Cluster Controller Application

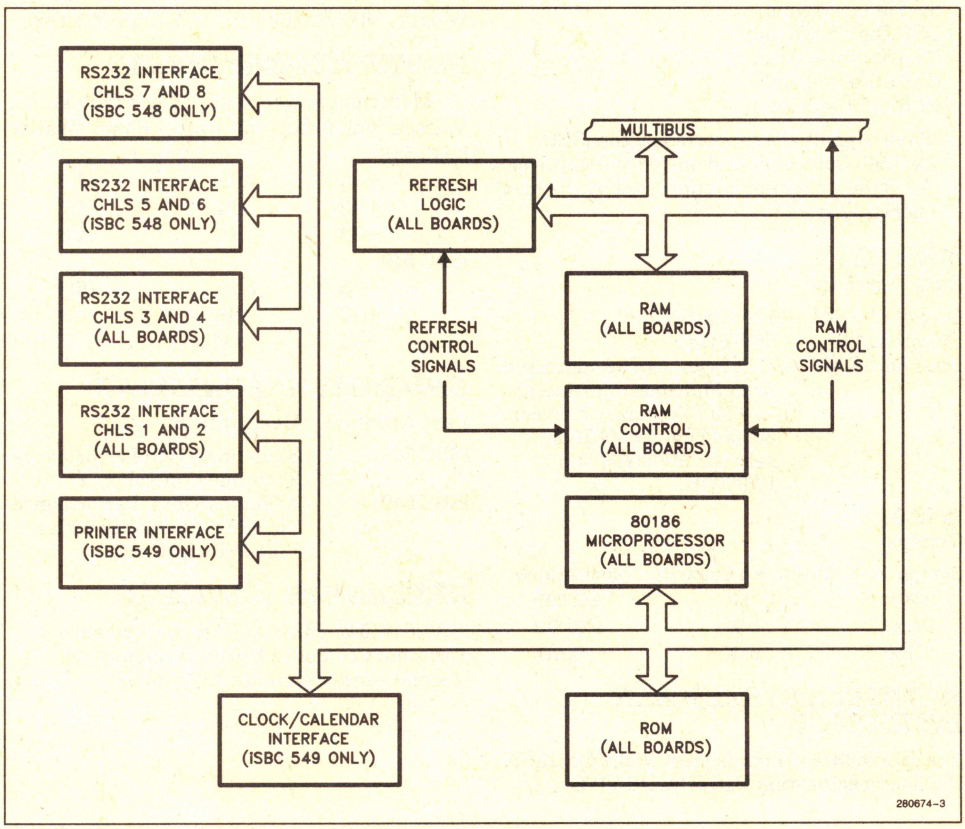


Figure 2: iSBC 548/549 Boards Block Diagram

intel

SPECIFICATIONS

SERIAL COMMUNICATIONS CHARACTERISTICS

Asynchronous only
- 6-8 bit character length
- 1, 1½, or 2 stop bits per character
- Parity
- Programmable clock
- Break Generation
- Framing error detection

Baud Rates

The on-board firmware can automatically detect and set baud rates of 150, 300, 600, 1200, 4800, 9600 and 19200. Other baud rates can be set by the host.

Serial RS232C Signals Supported
- CD Carrier Detect
- RXD Receive Data
- TXD Transmit Data
- DTR Data Terminal Ready
- SG Signal Ground
- DSR Data Set Ready
- RTS Ready to Send
- CTS Clear to Send
- RI Ringer Indicator

These signals are supported by the iSBC 548/549 Controller and on-board firmware. All signals may not be supported by the host operating system.

MEMORY

Dual Port RAM —

On-Board RAM — 128 Kbytes total
Private RAM — 96 Kbytes
Dual Port RAM — 32 Kbytes, can be addressed from Multibus interface at any 32K boundary between 80000H and F8000H or between F80000H and FF8000H.

EPROM Options —

Component	On-Board Capacity	Start Address
2764	16K	FC000H
27128	32K	F8000H
27256	64K	F000H

MULTIBUS SYSTEM BUS INTERFACE

The iSBC 548/549 boards meet Multibus (IEEE 796) bus specification D16 M24 I16 V0 E.

DEVICE DRIVERS

Check the latest release of the following operating systems for details:

iRMX 86 iRMX II

ENVIRONMENTAL CHARACTERISTICS

Temperature — 0°C to 55° C at 200 Linear Feet/Minute (LFM) Air Velocity

Humidity — 5% to 90% non-condensing (25°C to 70°C)

PHYSICAL CHARACTERISTICS

	iSBC 548	iSBC 549
Width	30.34cm (12.00 in)	30.34cm (12.00 in)
Length	16.87cm (6.75 in)	16.87cm (6.75 in)
Height	1.27 cm (.5 in)	1.27 cm (.5 in)
Weight	400 gm (14 oz)	358 gm (12.5 oz)

POWER REQUIREMENTS

Maximum Power Required per Voltage

Voltage (Volts)	Current (Amps)	Power (Watts)
iSBC 548		
+ 5	3.49	17.5
+12	.14	1.7
−12	.11	1.3
iSBC 549		
+ 5	3.26	16.3
+12	.07	.8
−12	.06	.7

ORDERING INFORMATION

Part Number	Description
iSBC 548	8 Channel High Performance Terminal Controller
iSBC 549	4 Channel High Performance Terminal Controller with Line Printer/Clock

REFERENCE MANUALS

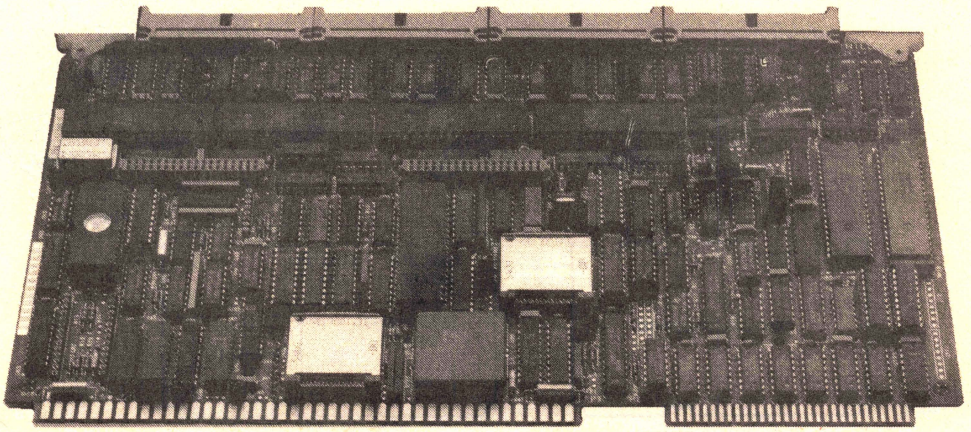
iSBC 546/547/548/549 High Performance Terminal Controller Hardware Reference Manual - Order Number 122704-002

iSBC 188/56
COMMUNICATING COMPUTER

- **iSBC Single Board Computer or Intelligent Slave Communication Board**
- **8 Serial Communications Channels, Expandable to 12 Channels on a Single Multibus Board**
- **8 MHz 80188 Microprocessor**
- **Supports RS232C Interface on 6 Channels, RS422A/449 or RS232C Interface Configurable on 2 Channels**
- **Supports Async, Bisync HDLC/SDLC, On-Chip Baud Rate Generation, Half/Full-Duplex, NRZ, NRZI or FM Encoding/Decoding**
- **7 On-Board DMA Channels for Serial I/O, 2 80188 DMA Channels for the iSBX Bus Interface**
- **Multibus Interface for System Expansion and Multimaster Configuration**
- **Two iSBX Connectors for Low Cost I/O Expansion**
- **256K Bytes Dual-Ported RAM On-Board**
- **Two 28-pin JEDEC PROM Sites Expandable to 6 Sites with the iSBC 341 MULTIMODULE Board for a Maximum of 192K Bytes EPROM**
- **Resident Firmware to Handle up to 12 RS232C Async Lines**

11

The iSBC 188/56 Communicating Computer is an intelligent 8-channel single board computer. This iSBC board adds the 8 MHz 80188 microprocessor-based communications flexibility to the Intel line of Multibus OEM microcomputer systems. Acting as a stand-alone CPU or intelligent slave for communication expansion, this board provides a high performance, low-cost solution for multi-user systems. The features of the iSBC 188/56 board are uniquely suited to manage higher-layer protocol requirements needed in today's data communications applications. This single board computer takes full advantage of Intel's VLSI technology to provide state-of-the-art, economic, computer based solutions for OEM communications-oriented applications.



280715-1

*IBM is a registered trademark of International Business Machines
*UNIX is a trademark of Bell Laboratories
*XENIX is a trademark of Microsoft Corporation

September 1993
Order Number: 280715-004

OPERATING ENVIRONMENT

The iSBC 188/56 board features have been designed to meet the needs of numerous communications applications. Typical applications include:

1. Terminal/cluster controller
2. Front-end processor
3. Stand-alone communicating computer

Terminal/Cluster Controller

A terminal/cluster controller concentrates communications in a central area of a system. Efficient handling of messages coming in or going out of the system requires sufficient buffer space to store messages and high speed I/O channels to transmit messages. More sophisticated applications, such as cluster controllers, also require character and format conversion capabilities to allow different types of terminals to be attached.

The iSBC 188/56 Communicating Computer is well suited for multi-terminal systems (see Figure 1). Up to 12 serial channels can be serviced in multi-user or cluster applications by adding two iSBX 354 MULTIMODULE boards. The dual-port RAM provides a large on-board buffer to handle incoming and outgoing messages at data rates up to 19.2K baud. Two channels are supported for continuous data rates greater than 19.2K baud. Each serial channel can be individually programmed for different baud rates to allow system configurations with differing terminal types. The firmware supplied on the iSBC 188/56 board supports up to 12 asynchronous RS232C serial channels, provides modem control and performs power-up diagnostics. The high performance of the on-board CPU provides intelligence to handle protocols and character handling typically assigned to the system CPU. The distribution of intelligence results in optimizing system performance by releasing the system CPU of routine tasks.

Front-End Processor

A front-end processor off-loads a system's central processor of tasks such as data manipulation and text editing of characters collected from the attached terminals. A variety of terminals require flexible terminal interfaces. Program code is often dynamically downloaded to the front-end processor from the system CPU. Downloading code requires sufficient memory space for protocol handling and program code. Flow control and efficient handling of interrupts require an efficient operating system to manage the hardware and software resources.

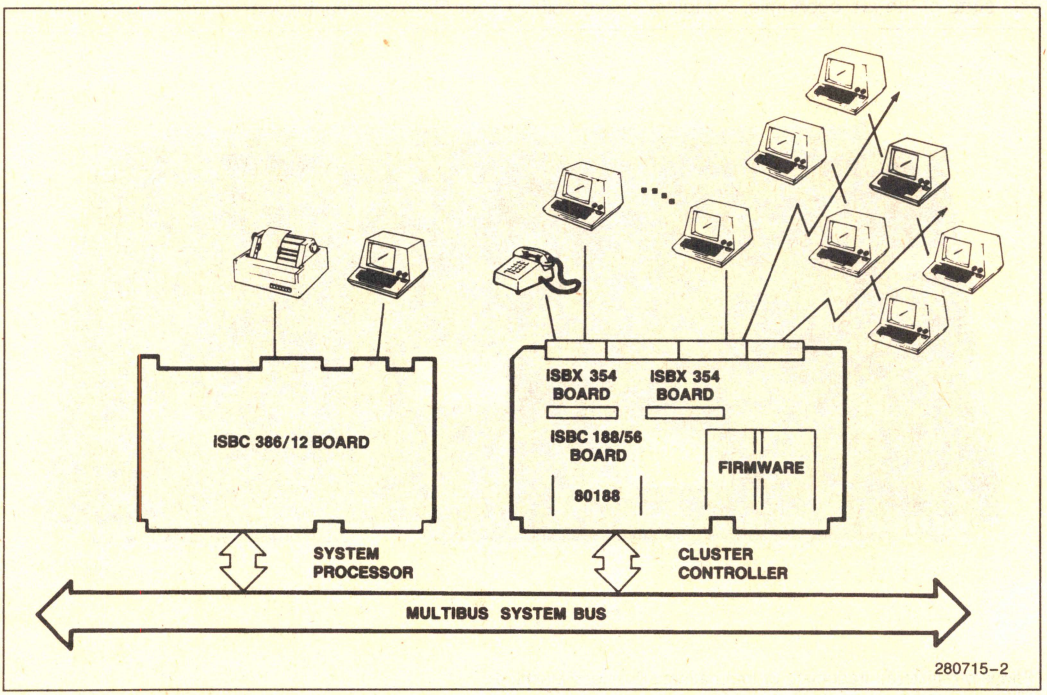


Figure 1. Terminal/Cluster Controller Application

The iSBC 188/56 board features are designed to provide a high performance solution for front-end processor applications (see Figure 2). A large amount of random access memory is provided for dynamic storage of program code. In addition, local memory sites are available for storing routine programs such as X.25, SNA or bisync protocol software. The serial channels can be configured for links to mainframe systems, point-to-point terminals, modems or multidrop configurations.

Stand-Alone Application

A stand-alone communication computer is a complete computer system. The CPU is capable of managing the resources required to meet the needs of multi-terminal, multi-protocol applications. These applications typically require multi-terminal support, floppy disk control, local memory allocation, and program execution and storage.

To support stand-alone applications, the iSBC 188/56 board uses the computational capabilities of an on-board CPU to provide a high-speed system solution controlling 8 to 12 channels of serial I/O (see Figure 3). The local memory available is large enough to handle special purpose code, execution code and routine protocol software. The Multibus interface can be used to access additional system functions. Floppy disk control and graphics capability can be added to the iSBC stand-alone computer through the iSBX connectors.

ARCHITECTURE

The four major functional areas are Serial I/O, CPU, Memory and DMA. These areas are illustrated in Figure 4.

Serial I/O

Eight HDLC/SDLC serial interfaces are provided on the iSBC 188/56 board. The serial interface can be expanded to 12 channels by adding 2 iSBX 354 MULTIMODULE boards. The HDLC/SDLC interface is compatible with IBM* system and terminal equipment and with CCITT's X.25 packet switching interface.

Four 82530 Serial Communications Controllers (SCC) provide eight channels of half/full duplex serial I/O. Six channels support RS232C interfaces. Two channels are RS232C/422/449 configurable and can be tri-stated to allow multidrop networks. The 82530 component is designed to satisfy several serial communications requirements; asynchronous,

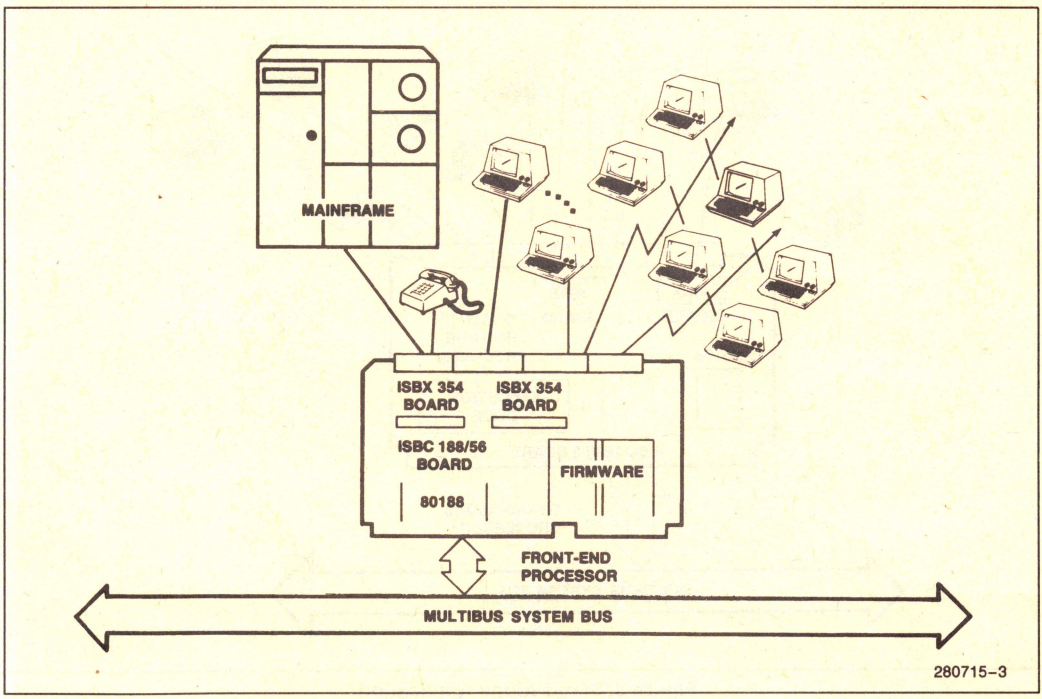


Figure 2. Front-End Processor Application

byte-oriented synchronous (HDLC/SDLC) modes. The increased capability at the serial controller point results in off-loading the CPU of tasks formerly assigned to the CPU or its associated hardware. Configurability of the 82530 allows the user to configure it to handle all asynchronous data formats regardless of data size, number of start or stop bits, or parity requirements. An on-chip baud rate generator allows independent baud rates on each channel.

The clock can be generated either internally with the SCC chip, with an external clock or via the NRZ1 clock encoding mechanism.

All eight channels can be configured as Data Terminal Equipment (DTE) or Data Communications Equipment (DCE). Table 1 lists the interfaces supported.

Table 1. iSBC 188/56 Interface Support

Connection	Synchronous	Asynchronous
	Modem to Direct	Modem to Direct
Point-to-Point	X** Channels	X Channels
Multidrop	0 and 1	0 and 1
Loop	X	N/A

**All 8 channels are denoted by X.

Central CPU

The 80188 central processor component provides high performance, flexibility and powerful processing. The 80188 component is a highly integrated microprocessor with an 8-bit data bus interface and a 16-bit internal architecture to give high performance. The 80188 is upward compatible with 86 and 186 software.

The 80188/82530 combination with on-board PROM/EPROM sites, and dual-port RAM provide the intelligence and speed to manage multi-user, multi-protocol communication operations.

Memory

There are two areas of memory on-board: dual-port RAM and universal site memory. The iSBC 188/56 board contains 256K bytes of dual-port RAM that is addressable by the 80188 on-board. The dual-port memory is configurable anywhere in a 16M byte address space on 64K byte boundaries as addressed from the Multibus port. Not all of the 256K bytes are visible from the Multibus bus side. The amount of dual-port memory visible to the Multibus side can

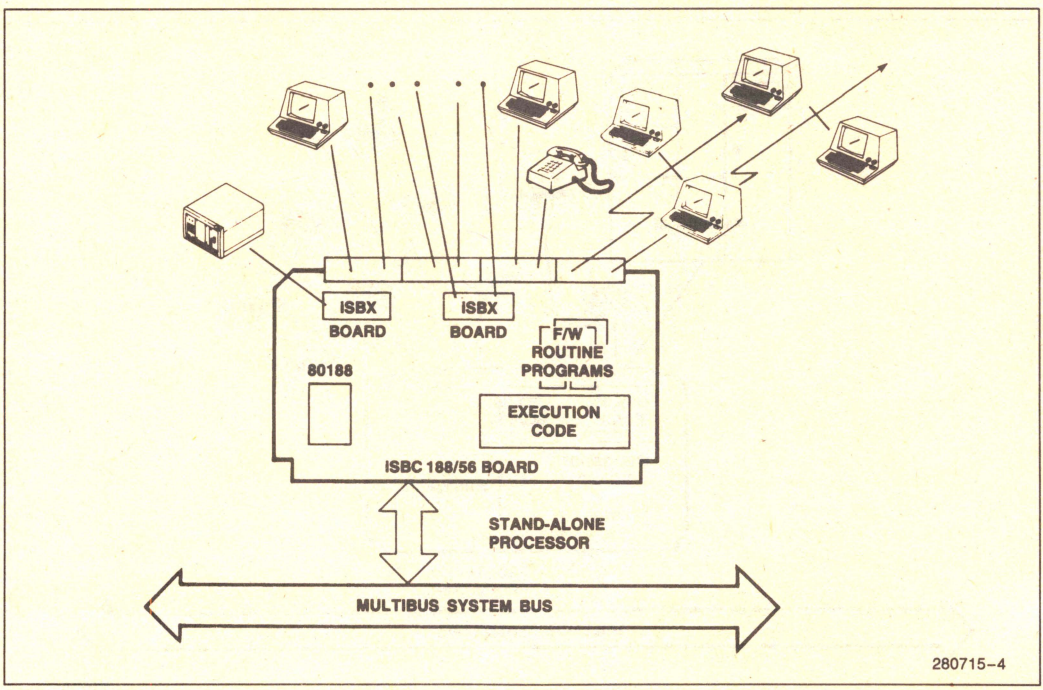


Figure 3. Stand-Alone Application

be set (with jumpers) to none, 16K bytes, or 48K bytes. In a multiprocessor system these features provide local memory for each processor and shared system memory configurations where the total system memory size can exceed one megabyte without addressing conflicts.

The second area of memory is universal site memory providing flexible memory expansion. Two 28-pin JEDEC sockets are provided. One of these sockets is used for the resident firmware as described in the FIRMWARE section.

The default configuration of the boards supports 16K byte EPROM devices such as the Intel 27128 component. However, these sockets can contain ROM, EPROM, Static RAM, or EEPROM. Both sockets must contain the same type of component (i.e. as the first socket contains an EPROM for the resident firmware, the second must also contain an EPROM with the same pinout). Up to 32K bytes can be addressed per socket giving a maximum universal site memory size of 64K bytes. By using the iSBC 341 MULTIMODULE board, a maximum of 192K bytes of universal site memory is available. This provides sufficient memory space for on-board network or resource management software.

On-Board DMA

Seven channels of Direct Memory Access (DMA) are provided between serial I/O and on-board dual port RAM by two 8237-5 components. Each of channels 0, 1, 2, 3, 5, 6, and 7 is supported by their own DMA line. Serial channels 0 and 1 are configurable for full duplex DMA. Configuring the full duplex DMA option for Channels 0 and 1 would require Channels 2 and 3 to be interrupt driven or polled. Channel 4 is interrupt driven or polled only.

Two DMA channels are integrated in the 80188 processor. These additional channels can be connected to the iSBX interfaces to provide DMA capability to iSBX MULTIMODULE boards such as the iSBX 218A Floppy Disk Controller MULTIMODULE board.

11

OPERATING SYSTEM SUPPORT

Intel offers run-time foundation software to support applications that range from general purpose to high-performance solutions.

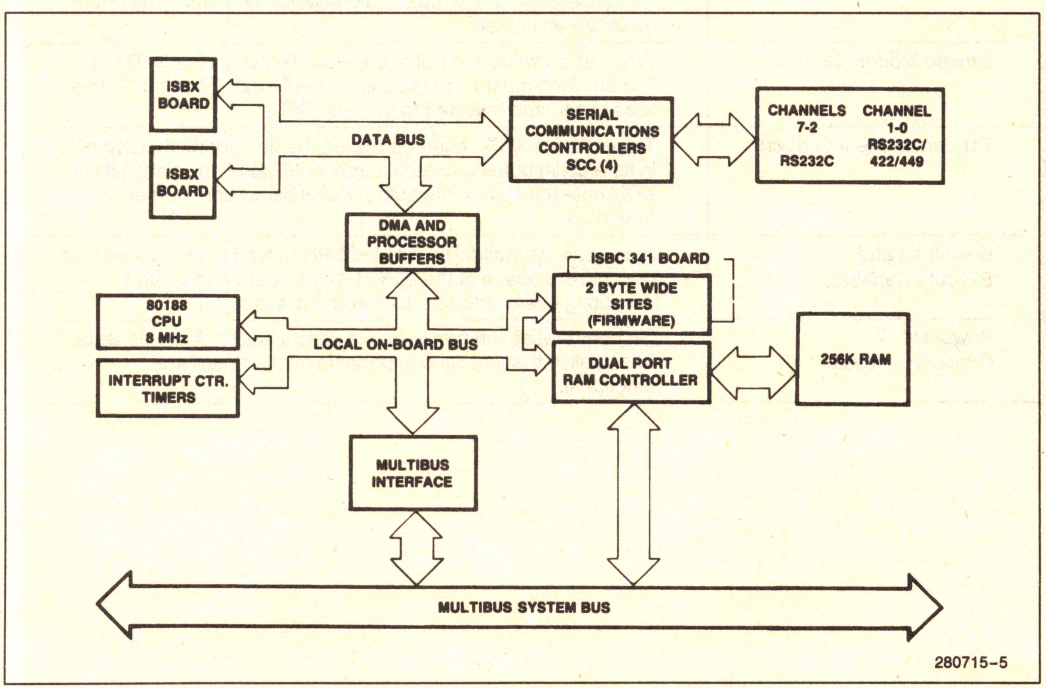


Figure 4. Block Diagram of iSBC 188/56 Board

The iRMX Operating System provides a rich set of features and options to support sophisticated stand-alone communications applications on the iSBC 188/56 Communicating Computer. If the iSBC 188/56 board is acting as an intelligent slave in a system environment, an iRMX driver resident in the host CPU is available.

The System Debug Monitor (SDM) supports target system debugging for the iSBC 188/56 Advanced Communicating Commputer board. The monitor contains the necessary hardware, software and documentation required to interface the iSBC 188/56 target system to an Intel microcomputer development system for debugging application software.

FIRMWARE

The iSBC 188/56 Communicating Commputer board is supplied with resident firmware that supports up to 12 RS232C asynchronous serial channels. In addition, the firmware provides a facility for a host CPU to download and execute code on the iSBC 188/56 board. Simple power-up confidence tests are also included to provide a quick diagnostic service. The firmware converts the iSBC 188/56 Commputer board to a slave communications controller. As a slave communications controller, it requires a separate Multibus host CPU board and requires the use of Multibus interrupt line to signal the host processor. Table 2 summarizes the features of the firmware.

Table 2. Features of the iSBC 188/56 Firmware

Feature	Description
Asynchronous Serial Channel Support	Supports the serial channels in asynchronous ASCII mode. Parameters such as baud rate, parity generation, parity checking and character length can be programmed independently for each channel.
Block Data Transfer (On Output)	Relieves the host CPU of character-at-a-time interrupt processing. The iSBC 188/56 board accepts blocks of data for transmission and interrupts the processor only when the entire block is transmitted.
Limited Modem Control	Provides software control of the Data Terminal Ready (DTR) line on all channels. Transitions on the Carrier Detect (CD) line are sensed and reported to the host CPU.
Tandem Modem Support	Transmits an XOFF character when the number of characters in its receive buffer exceeds a threshold value and transmits an XON character when the buffer drains below some other threshold.
Download and Execute Capability	Provides a capability for the host CPU to load code anywhere in the address space of the iSBC 188/56 board and to start executing at any address in its address space.
Power Up Confidence Tests	On board reset, the firmware executes a series of simple tests to establish that crucial components on the board are functional.

INTERRUPT CAPABILITY

The iSBC 188/56 board has two programmable interrupt controllers (PICs). One is integrated into the 80188 processor and the other in the 80130 component. The two controllers are configured with the 80130 controller as the master and the 80188 controller as the slave. Two of the 80130 interrupt inputs are connected to the 82530 serial controller components to provide vector interrupt capabilities by the serial controllers. The iSBC 188/56 board provides 22 interrupt levels. The highest level is the NMI (Non-Maskable Interrupt) line which is directly tied to the 80188 CPU. This interrupt is typically used for signaling catastrophic events (e.g. power failure).

There are 5 levels of interrupts internal to the 80188 processor. Another 8 levels of interrupts are available from the 80130 component. Of these 8, one is tied to the programmable interrupt controller (PIC) of the 80188 CPU. An additional 8 levels of interrupts are available at the Multibus interface. The iSBC 188/56 board does not support bus vectored interrupts. Table 3 lists the possible interrupt sources.

Table 3. Interrupt Request Sources

Device	Function	Number of Interrupts
Multibus Interface INT0–INT7	Requests from Multibus resident peripherals or other CPU boards.	8
82530 Serial Controllers	Transmit buffer empty, receive buffer full and channel errors 1 and external status.	8 per 82530 Total = 32
Internal 80188 Timer and DMA	Timer 0, 1, 2 outputs and 2 DMA channel interrupts.	5
80130 Timer Outputs	Timer 0, 1, 2 outputs of 80130.	3
Interrupt from Flag Byte Logic	Flag byte interrupt set by Multibus master (through Multibus I/O Write).	1
Bus Flag Interrupt	Interrupt to Multibus (Selectable for INT0 to INT7) generated from on-board 80188 I/O Write.	1
SBX Connectors SBX DMA	Function determined by iSBX board. DMA interrupt from iSBX (TDMA).	4 (Two per Connector) 2
Bus Fail-Safe Timeout Interrupt.	Indicates iSBC 188/48 board timed out either waiting for Multibus access or timed out from no acknowledge while on Multibus System Bus.	1
Latched Interrupt	Converts pulsed event to a level interrupt. Example: 8237A-5 EOP.	1
OR-Gate Matrix	Concentrates up to 4 interrupts to 1 interrupt (selectable by stake pins).	1
Ring Indicator Interrupt	Latches a ring indicator event from serial channels 4, 5, 6, or 7.	1
NOR-Gate Matrix	Inverts up to 2 interrupts into 1 (selectable by stake pins).	1

EXPANSION

EPROM Expansion

Memory may be expanded by adding Intel compatible memory expansion boards. The universal site memory can be expanded to six sockets by adding the iSBC 341 MULTIMODULE board for a maximum total of 192K bytes of universal site memory.

iSBX I/O Expansion Bus (IEEE 959)

Two 8-bit iSBX bus connectors are provided on the iSBC 188/56 board. Using iSBX modules additional functions can be added to extend the I/O capability of the board. In addition to specialized or custom designed iSBX boards, there is a broad range of iSBX boards from Intel including serial and parallel I/O, analog I/O, and IEEE 488 GPIB, boards.

The serial I/O SBX boards available include the iSBX 354 Dual Channel Expansion MULTIMODULE board. Each iSBX 354 MULTIMODULE board adds two channels of serial I/O to the iSBC 188/56 board for a maximum of twelve serial channels. The 82530 serial communications controller on the MULTIMODULE board handles a large variety of serial communications protocols. This is the same serial controller as is used on the iSBC 188/56 board to offer directly compatible expansion capability for the iSBC 188/56 board.

MULTIBUS INTERFACE

The iSBC 188/56 board can be a Multibus master or intelligent slave in a multimaster system. The iSBC 188/56 board incorporates a flag byte signalling mechanism for use in multiprocessor environments where the iSBC 188/56 board is acting as an intelligent slave. The mechanism provides an interrupt handshake from the Multibus System Bus to the on-board-processor and vice-versa.

The Multimaster capabilities of the iSBC 188/56 board offers easy expansion of processing capacity and the benefits of multiprocessing. Memory and I/O capacity may be expanded and additional functions added using Intel Multibus compatible expansion boards.

SPECIFICATIONS

Word Size

Instruction—8, 16, 24 or 32 bits
Data Path—8 bits

Processor Clock	82530 Clock	DMA Clock
8 MHz	4.9152 MHz	4 MHz

Dual Port RAM

iSBC 188/56 Board—256 bytes

As viewed from the 80188—64K bytes

As viewed from the Multibus System Bus—Choice: 0, 16K or 48K

EPROM

iSBC 188/56 Board Using:	Size	On Board Capacity	Address Range
2732	4K	8K bytes	FE000–FFFFF$_H$
2764	8K	16K bytes	FC000–FFFFF$_H$
27128	16K	32K bytes	F8000–FFFFF$_H$
27256	32K	64K bytes	F0000–FFFFF$_H$
27512	64K	128K bytes	E0000–FFFFF$_H$

Memory Expansion

EPROM with iSBC 341 Board Using:	Capacity	Address Range
2732	24K bytes	F8000–FFFFF$_H$
2764	48K bytes	F0000–FFFFF$_H$
27128	96K bytes	E0000–FFFFF$_H$
27256	192K bytes	C0000–FFFFF$_H$

I/O Capacity

Serial—8 programmable lines using four 82530 components

SBX Bus—2 SBX single-wide boards

Serial Communications Characteristics

Synchronous—Internal or external character synchronization on one or two synchronous characters

Asynchronous—5–8 bits and 1, 1½, or 2 stop bits per character; programmable clock factor; break detection and generation; parity, overrun, and framing error detection.

Baud Rates

Synchronous X1 Clock	
Baud Rate	**82530 Count Value (Decimal)**
64000	36
48000	49
19200	126
9600	254
4800	510
2400	1022
1800	1363
1200	2046
300	8190
Asynchronous X16 Clock	
Baud Rate	**82530 Count Value (Decimal)**
19200	6
9600	14
4800	30
2400	62
1800	83
1200	126
300	510
110	1394

Interfaces

iSBX BUS

The iSBC 188/56 board meets IEEE 959 compliance level D8/8 DMA

MULTIBUS SYSTEM BUS

The iSBC 188/56 board meets IEEE 796 compliance level Master/Slave D8 M24 I16 VO EL.

SERIAL RS232C SIGNALS

CD	Carrier
CTS	Clear to Send
DSR	Data Set Ready
DTE TXC	Transmit Clock
DTR	Data Terminal Ready
RTS	Request to Send
RXC	Receive Clock
RXD	Receive Data
SG	Signal Ground
TXD	Transmit Data
RI	Ring Indicator

RS422A/449 SIGNALS

RC	Receive Common
RD	Receive Data
RT	Receive Timing
SD	Send Data
TT	Terminal Timing

11

Environmental Characteristics

Temperature: 0 to 55°C at 200 Linear Feet/Min. (LFM) Air Velocity

Humidity: to 90%, non-condensing (25°C to 70°C)

Physical Characteristics

Width: 30.48 cm (12.00 in)
Length: 17.15 cm (6.75 in)
Height: 1.04 cm (0.41 in)
Weight: 595 gm (21 oz)

Electrical Characteristics

The power required per voltage for the iSBC 188/56 board is shown below. These numbers do not include the current required by universal memory sites or expansion modules.

Voltage (Volts)	Current (Amps) typ.	Power (Watts) typ.
+5	4.56A	22.8W
+12	0.12A	1.5W
−12	0.11A	1.3W

Reference Manual

iSBC 188/56 Advanced Data Communications Computer Reference Manual Order Number 148209.

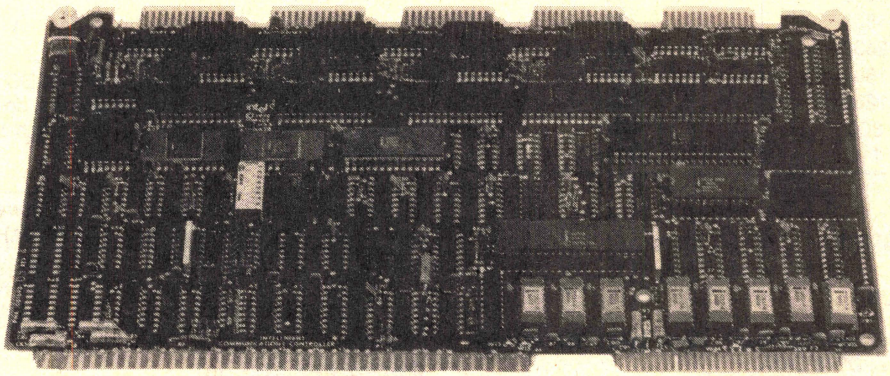
ORDERING INFORMATION

Order Code	Description
SBC188/56	8-Serial Channel Advanced Communicating Computer

iSBC 544A
INTELLIGENT COMMUNICATIONS CONTROLLER

- **iSBC Communications Controller Acting as a Single Board Communications Computer or an Intelligent Slave for Communications Expansion**
- **On-Board Dedicated 8085A Microprocessor Providing Communications Control and Buffer Management for Four Programmable Synchronous/Asynchronous Channels**
- **Sockets for Up To 8K Bytes of EPROM**
- **16K Bytes of Dual Port Dynamic Read/Write Memory with On-Board Refresh**
- **Extended Multibus Addressing Permits Board Partitioning into 16K-Byte Segments in a 1-Megabyte Address Space**
- **Ten Programmable Parallel I/O Lines Compatible with Bell 801 Automatic Calling Unit**
- **Twelve Levels of Programmable Interrupt Control**
- **Individual Software Programmable Baud Rate Generation for Each Serial I/O Channel**
- **Three Independent Programmable Interval Timer/Counters**
- **Interface Control for Auto Answer and Auto Originate Modem**

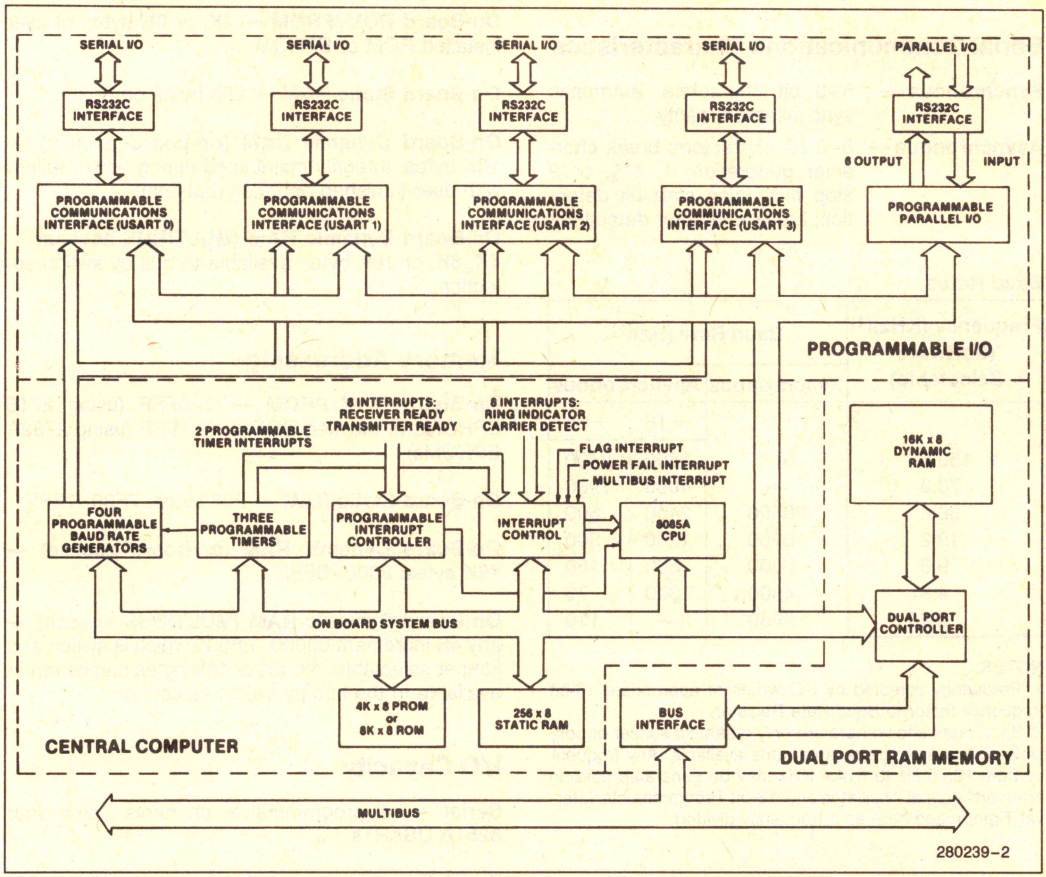
The iSBC 544A Intelligent Communications Controller is a member of Intel's family of Multibus single-board computers, memory, I/O, and peripheral controller boards. The iSBC 544A board is a complete communications controller. The on-board 8085A CPU may perform local communications processing by directly interfacing with on-board read/write memory, nonvolatile read only memory, four synchronous/asynchronous serial I/O ports, RS232/RS366 compatible parallel I/O, programmable timers, and programmable interrupts.



280239-1

August 1993
Order Number: 280239-002

BLOCK DIAGRAM



iSBC 544A Intelligent Communications Controller Block Diagram

11

SPECIFICATIONS

Serial Communications Characteristics

Synchronous — 5–8 bit characters; automatic sync insertion; parity.

Asynchronous — 5–8 bit characters; break character generation; 1, 1½, or 2 stop bits; false start bit detection; break character detection.

Baud Rates

Frequency (KHz)(1) (Software Selectable)	Baud Rate (Hz)(2)		
	Synchronous	Asynchronous	
		÷16	÷64
153.6	—	9600	2400
76.8	—	4800	1200
38.4	38400	2400	600
19.2	19200	1200	300
9.6	9600	600	150
4.8	4800	300	75
6.98	6980	—	110

NOTES:
1. Frequency selected by I/O writes of appropriate 16-bit frequency factor to Baud Rate Register.
2. Baud rates shown here are only a sample subset of possible software programmable rates available. Any frequency from 18.75 Hz to 614.4 KHz may be generated utilizing on-board crystal oscillator and 16-bit Programmable Interval Timer (used here as a frequency divider).

8085A CPU

Word Size — 8, 16 or 24 bits/instruction; 8 bits of data

Cycle Time — 1.45/μs ±0.01% for fastest executable instruction; i.e., four clock cycles.

Clock Rate — 2.76 MHz ± 0.1%

System Access Time

Dual port memory — 740 ns

NOTE:
Assumes no refresh contention.

Memory Capacity

On-Board ROM/PROM — 4K, or 8K bytes of user installed ROM or EPROM

On-Board Static RAM — 256 bytes on 8155

On-Board Dynamic RAM (on-board access) — 16K bytes. Integrity maintained during power failure with user-furnished batteries (optional)

On-Board Dynamic RAM (MULTIBUS access) — 4K, 8K, or 16K bytes available to bus by swtich selection

Memory Addressing

On-Board ROM/PROM — 0–0FFF (using 2716 EPROMs or masked ROMs); 0–1FFF (using 2732A EPROMs)

On-Board Static RAM — 256 bytes: 7F00–7FFF

On-Board Dynamic RAM (on-board access) — 16K bytes: 8000–BFFF.

On-Board Dynamic RAM (MULTIBUS access) — any 4K increment 00000–FF000 which is switch and jumper selectable. 4K, 8K or 16K bytes can be made available to the bus by switch selection.

I/O Capacity

Serial — 4 programmable channels using four 8251A USARTs

Parallel — 10 programmable lines available for Bell 801 ACU, or equivalent use. Two auxiliary jumper selectable signals

I/O Addressing

On-Board Programmable I/O

Port	Data	Control
USART 0	D0	D1
USART 1	D2	D3
USART 2	D4	D5
USART 3	D6	D7
8155 PPI	E9 (Port A) EA (Port B) EB (Port C)	E8

Interrupts

Address for 8259A Registers (Hex notation, I/O address space)

E6 Interrupt request register
E6 In-service register
E7 Mask register
E6 Command register
E7 Block address register
E6 Status (polling register)

NOTE:
Several registers have the same physical address: Sequence of access and one data bit of the control word determines which register will respond.

Interrupt levels routed to the 8085 CPU automatically vector the processor to unique memory locations:

24 TRAP
3C RST 7.5
34 RST 6.5
2C RST 5.5

Timers

Addresses for 8253 Registers (Hex notation, I/O address space)

Programmable Interrupt Timer One

D8	Timer 0	BDG0
D9	Timer 1	BDG1
DA	Timer 2	BDG2
DB	Control register	

Programmable Interrupt Timer Two

DC	Timer 0	BDG3
DD	Timer 1	BDG4
DE	Timer 2	TINT1
DF	Control register	

Address for 8155 Programmable Timer

E8	Control	
	Timer (LSB)	TINT0
ED	Timer (MSB)	TINT0

Input Frequencies — Jumper selectable reference 1.2288 MHz ± 0.1% (0.814 μs period nominal) or 1.843 MHz ±0.1% crystal (0.542 μs period, nominal)

11

Output Frequencies (at 1.2288 MHz)

Function	Single Timer/Counter		Dual Timer/Counter (two timers cascaded)	
	Min	Max	Min	Max
Real-Time Interrupt Interval	1.63 μs	53.3 μs	3.26 μs	58.25 min
Rate Generator (frequency)	18.75 Hz	614.4 KHz	0.00029 Hz	307.2 KHz

Interfaces

Serial I/O — EIA Standard RS232C signals provided and supported:

Carrier Detect	Receiver Data
Clear to Send	Ring Indicator
Data Set Ready	Secondary Receive Data*
Data Terminal Ready	Secondary Transmit Data *
Request to Send	Transmit Clock
Receive Clock	Transmit Data
	DTE Transmit clock

* Optional if parallel I/O port is not used as Automatic Calling Unit.

Parallel I/O — Four inputs and eight outputs (includes two jumper selectable auxiliary outputs). All signals compatible with EIA Standard RS232C. Directly compatible with Bell Model 801 Automatic Calling Unit, or equivalent.

On-Board Addressing

All communications to the parallel and serial I/O ports, to the timers, and to the interrupt controller, are via read and write commands from the on-board 8085A CPU.

Auxiliary Power

An auxiliary power bus is provided to allow separate power to RAM for systems requiring battery backup of read/write memory. Selection of this auxiliary RAM power bus is made via jumpers on the board.

Memory Protect

An active-low TTL compatible memory protect signal is brought out on the auxiliary connector which, when asserted, disables read/write access to RAM memory on the board. This input is provided for the protection of RAM contents during the system power-down sequences.

Bus Drivers

Function	Characteristic	Sink Current (mA)
Data	Tri-state	50
Address	Tri-state	15
Commands	Tri-state	32

NOTE:
Used as a master in the single board communications computer mode.

Physical Characteristics

Width:	30.48 cm (12.00 inches)
Depth:	17.15 cm (6.75 inches)
Thickness:	1.27 cm (0.50 inch)
Weight:	3.97 gm (14 ounces)

Electrical Characteristics

DC Power Requirements

Current Requirements				
Configuration	**V_{CC} = +5V ±5% (max)**	**V_{DD} = ±12V ±5% (max)**	**V_{BB} = −5V(3) ±5% (max)**	**V_{AA} = −12V ±5% (max)**
With 4K EPROM (using 2716)	I_{CC} = 3.4A max	I_{DD} = 350 mA max	I_{BB} = 5 mA max	I_{AA} = 200 mA max
Without EPROM	3.3A max	350 mA max	5 mA max	200 mA max
RAM only(1)	390 mA max	176 mA max	5 mA max	—
RAM(2) refresh only	390 mA max	20 mA max	5 mA max	

NOTES:
1. For operational RAM only, for AUX power supply rating.
2. For RAM refresh only. Used for battery backup requirements. No RAM accessed.
3. V_{BB} is normally derived on-board from V_{AA}, eliminating the need for a V_{BB} supply. If it is desired to supply V_{BB} from the bus, the current requirement is as shown.

Environmental Characteristics

Operating Temperature: 0°C to 55°C (32°F to 131°F)

Relative Humidity: To 90% without condensation

ORDERING INFORMATION

Order Code	Description
SBC 544A	Intelligent Communications Controller

Reference Manual

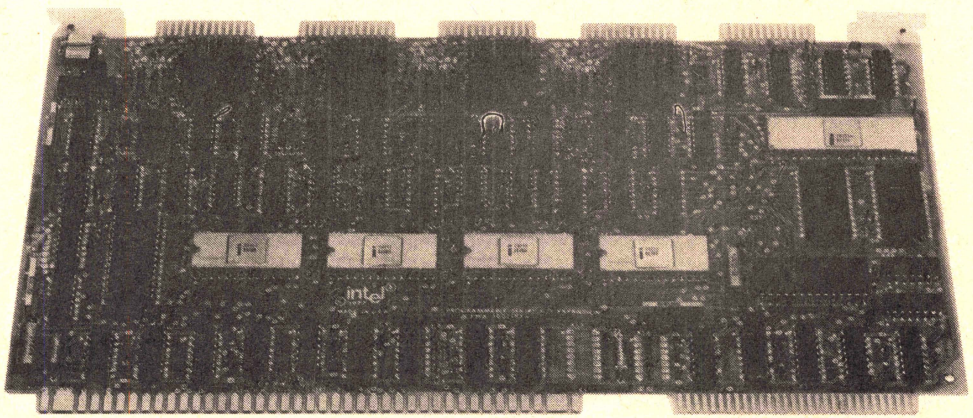
502160 — iSBC 544 Intelligent Communications Controller Board Hardware Reference Manual (NOT SUPPLIED)

intel

iSBC 534
FOUR CHANNEL COMMUNICATION EXPANSION BOARD

- **Serial I/O Expansion Through Four Programmable Synchronous and Asynchronous Communications Channels**
- **Individual Software Programmable Baud Rate Generation for Each Serial I/O Channel**
- **Two Independent Progammable 16-Bit Interval Timers**
- **Sixteen Maskable Interrupt Request Lines with Priority Encoded and Programmable Interrupt Algorithms**
- **Jumper Selectable Interface Register Addresses**
- **16-Bit Parallel I/O Interface Compatible with Bell 801 Automatic Calling Unit**
- **RS232C/CCITT V.24 Interfaces Plus 20 mA Optically Isolated Current Loop Interfaces (Sockets)**
- **Programmable Digital Loopback for Diagnostics**
- **Interface Control for Auto Answer and Auto Originate Modems**

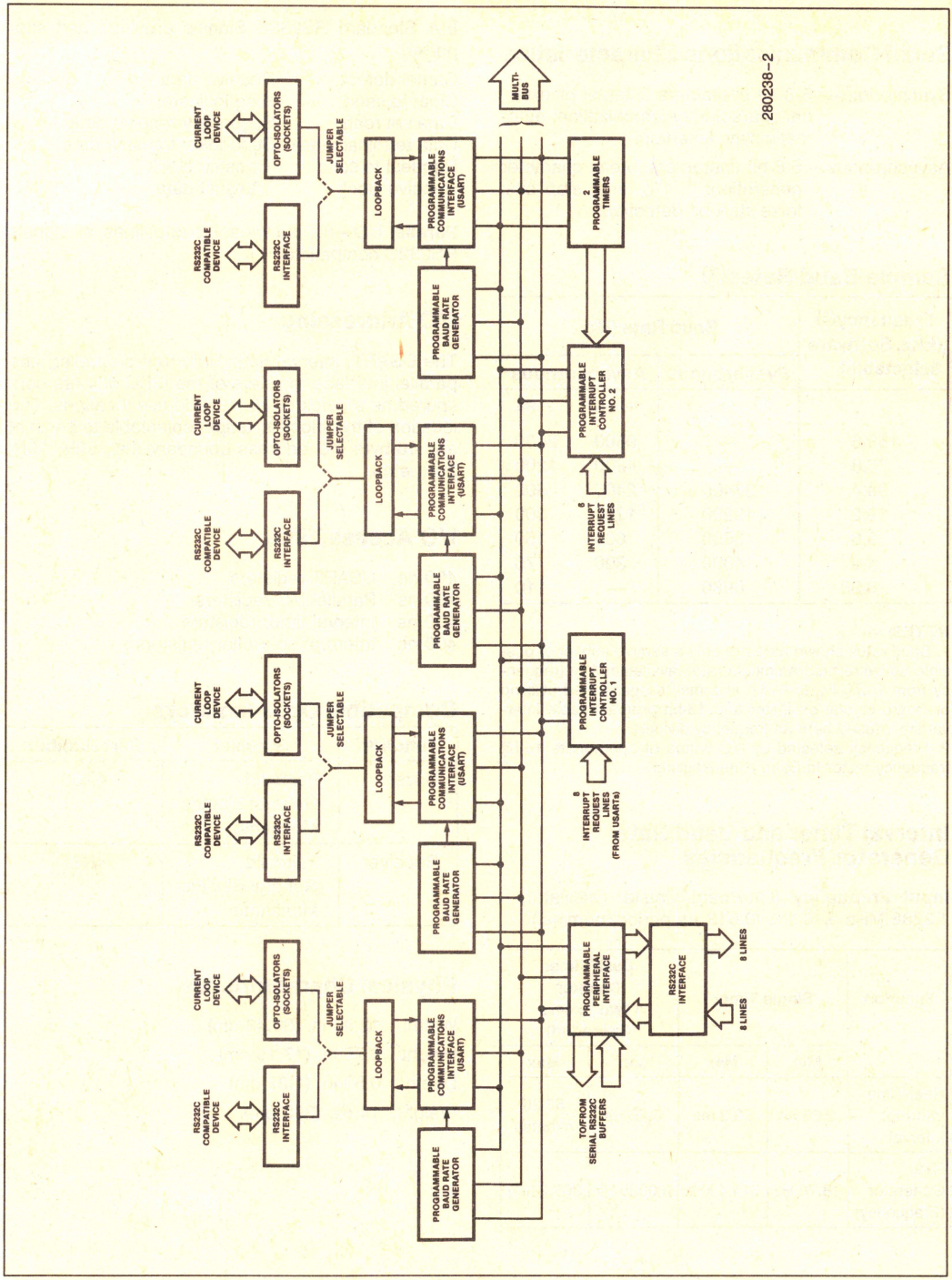
The iSBC 534 Four Channel Communication Expansion Board is a member of Intel's complete line of memory and I/O expansion boards. The iSBC 534 interfaces directly to any single board computer via the Multibus to provide expansion of system serial communications capability. Four fully programmable synchronous and asynchronous serial channels with RS232C buffering and provision for 20 mA optically isolated current loop buffering are provided. Baud rates, data formats, and interrupt priorities for each channel are individually software selectable. In addition to the extensive complement of EIA Standard RS232C signals provided, the iSBC 534 provides 16 lines of RS232C buffered programmable parallel I/O. This interface is configured to be directly compatible with the Bell Model 801 automatic calling unit. These capabilities provide a flexible and easy means for interfacing Intel iSBC based systems to RS232C and optically isolated current loop compatible terminals, cassettes, asynchronous and synchronous modems, and distributed processing networks.



280238-1

August 1993
Order Number: 280238-002

BLOCK DIAGRAM



iSBC 534 Four Channel Communications Expansion Board Block Diagram

SPECIFICATIONS

Serial Communications Characteristics

Synchronous— 5-8 bit characters; internal or external character synchronization; automatic sync insertion.

Asynchronous— 5-8 bit characters; break character generation; 1, 1½, or 2 stop bits; false start bit detection.

Sample Baud Rates(1)

Frequency(2) (kHz, Software Selectable)	Baud Rate (Hz)		
	Synchronous	Asynchronous	
		÷ 16	÷ 64
153.6	—	9600	2400
76.8	—	4800	1200
38.4	38400	2400	600
19.2	19200	1200	300
9.6	9600	600	150
4.8	4800	300	75
6.98	6980	—	110

NOTES:
1. Baud rates shown here are only a sample subset of possible software programmable rates available. Any frequency from 18.75 Hz to 614.4 kHz may be generated utilizing on-board crystal oscillator and 16-bit programmable interval timer (used here as frequency divider).
2. Frequency selected by I/O writes of appropriate 16-bit frequency factor to Baud Rate Register.

Interval Timer and Baud Rate Generator Frequencies

Input Frequency (On-Board Crystal Oscillator)— 1.2288 MHz ± 0.1% (0.813 μs period, nominal)

Function	Single Timer		Dual/Timer Counter (Two Timers Cascaded)	
	Min	Max	Min	Max
Real-Time Interrupt Interval	1.63 μs	53.3 ms	3.26 μs	58.25 minutes
Rate Generator (Frequency)	18.75 Hz	614.4 kHz	0.0029 Hz	307.2 kHz

Interfaces—RS232C Interfaces

EIA Standard RS232C Signals provided and supported:

Carrier detect	Receive data
Clear to send	Ring indicator
Data set ready	Secondary receive data
Data terminal ready	Secondary transmit data
Request to send	Transmit clock
Receive clock	Transmit data

Parallel I/O—8 input lines, 8 output lines, all signals RS232C compatible

I/O Addressing

The USART, interval timer, interrupt controller, and parallel interface registers of the iSBC 534 are configured as a block of 16 I/O address locations. The location of this block is jumper selectable to begin at any 16-byte I/O address boundary (i.e., 00H, 10H, 20H, etc.).

I/O Access Time

400 ns	USART registers
400 ns	Parallel I/O registers
400 ns	Interval timer registers
400 ns	Interrupt controller registers

Compatible Opto-Isolators

Function	Supplier	Part Number
Driver	Fairchild General Electric Monsanto	4N33
Receiver	Fairchild General Electric Monsanto	4N37

Physical Characteristics

Width: 12.00 in. (30.48 cm)
Height: 6.75 in. (17.15 cm)
Depth: 0.50 in. (1.27 cm)
Weight: 14 oz. (398 gm)

Electrical Characteristics

Average DC Current

Voltage	Without Opto-Isolators	With Opto-Isolators(1)
$V_{CC} = +5V$	1.9 A, max	1.9 A, max
$V_{DD} = +12V$	275 mA, max	420 mA, max
$V_{AA} = -12V$	250 mA, max	400 mA, max

NOTE:
1. With four 4N33 and four 4N37 opto-isolator packages installed in sockets provided to implement four 20 mA current loop interfaces.

Environmental Characteristics

Operating Temperature: 0°C to +55°C

Reference Manual

502140—iSBC 534 Hardware Reference Manual (NOT SUPPLIED)

ORDERING INFORMATION

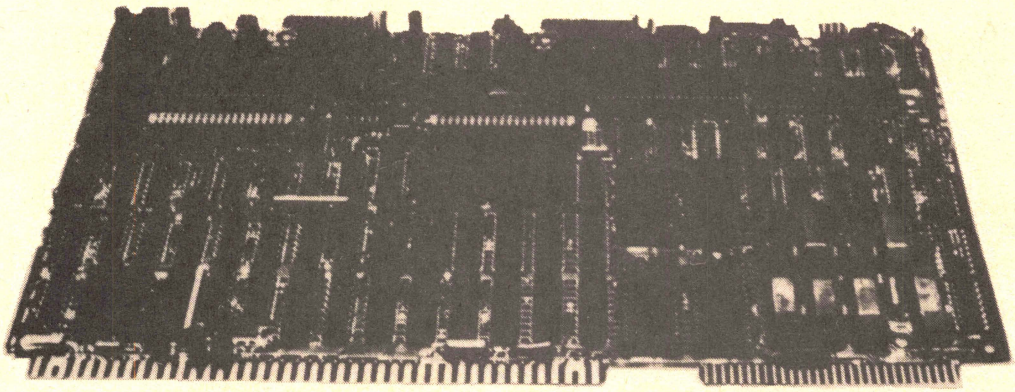
Order Code	Description
SBC534	Four Channel Communication Expansion Board

11

iSBC 88/45
DATA COMMUNICATIONS PROCESSOR BOARD

- Three HDLC/SDLC Half/Full-Duplex Communication Channels—Optional ASYNC/SYNC on Two Channels
- Supports RS232C (Including Modem Support), CCITT V.24, or RS422A/449 Interfaces
- On-Board DMA Supports 800K Baud Operation
- Self-Clocking NRZI SDLC Loop Data Link Interface
  - — Point-to-Point
  - — Multidrop
- Software Programmable Baud Rate Generation
- 8088 (8088-2) Microprocessor Operates at 8 MHz
- iSBC 337A Numeric Data Processor Option Supported
- 16K Bytes Static RAM (12K Bytes Dual-Ported)
- Four 28-Pin JEDEC Sites for EPROM/RAM Expansion; Four Additional 28-Pin JEDEC Sites Added with iSBC 341 Board
- Two SBX Bus (IEEE 959) Connectors
- Multibus Interface Supports Multimaster Configuration

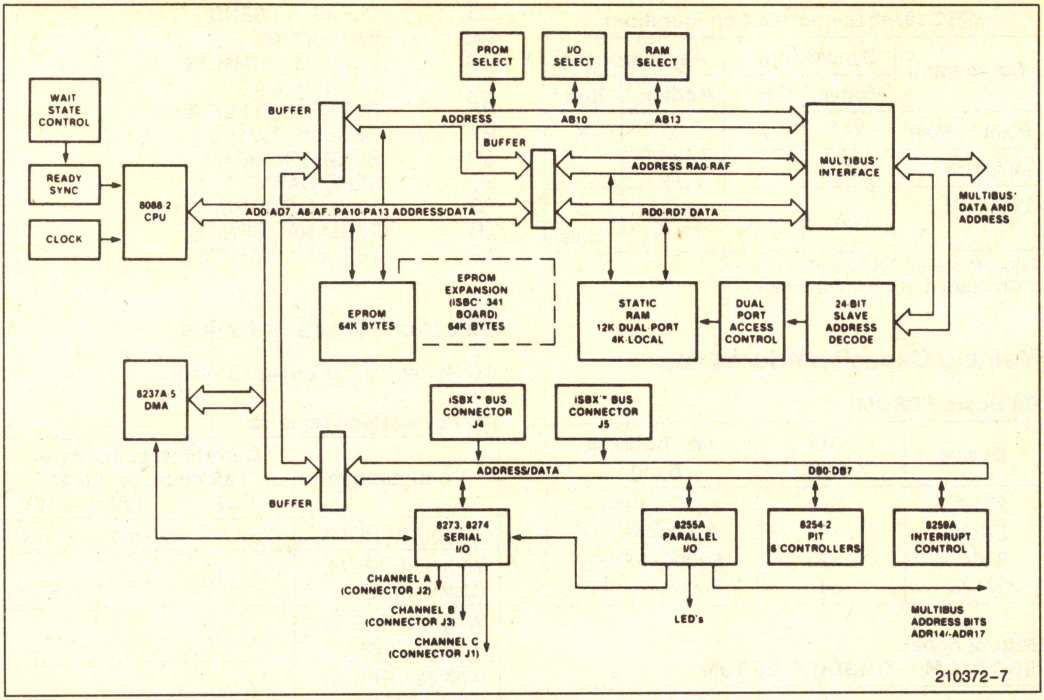
The iSBC 88/45 Data Communications Processor (DCP) Board adds 8 MHz, 8088 (8088-2) 8-bit microprocessor-based communications flexibility to the Intel line of Multibus OEM microcomputer systems. The iSBC 88/45 DCP board offers asynchronous, synchronous, SDLC, and HDLC serial interfaces for gateway networking or general purpose solutions. The iSBC 88/45 DCP board provides the CPU, system clock, EPROM/RAM, serial I/O ports, priority interrupt logic, and programmable timers to facilitate higher-level application solutions.



210372-1

August 1993
Order Number: 210372-003

BLOCK DIAGRAM



Block Diagram of the iSBC 88/45 DCP Board

SPECIFICATIONS

Word Size

Instruction: 8, 16, 24, or 32 bits
Data: 8 or 16 bits

System Clock

8 MHz: ±0.1%

NOTE:
Jumper selectable for 4 MHz operation with iSBC 337 Numeric Data Processor module or ICE-88 product.

Cycle Time

Basic Instruction Cycle at 8.00 MHz: 1.25 μs, 250 ns (assumes instruction in the queue)

NOTE:
Basic instruction cycle is defined as the fastest instruction time (i.e., two clock cycles).

Memory Cycle Time

RAM: 500 ns (no wait states)
EPROM: jumper selectable from 500 ns to 625 ns.

On-Board RAM*

K Bytes	Hex Address Range
16 (total) 12 (dual-ported)	0000–3FFF 1000–3FFF

*Four iSBC 88/45 EPROM sockets support JEDEC 24/28-pin standard EPROMs and RAMs (3 sockets); iSBC 341 (4 sockets)

Environmental Characteristics

Temperature: 0°C to +55°C, free moving air across the base board and MULTIMODULE board

Humidity: 90%, non-condensing

Physical Characteristics

Width: 30.48 cm (12.00 in)
Length: 17.15 cm (6.75 in)
Height: 1.50 cm (0.59 in)
Weight: 6.20 gm (22 oz)

Configurations

iSBC 88/45 Supported Configurations

Connection	Synchronous		Asynchronous	
	Modem	Direct	Modem*	Direct
Point-to-Point	X**	X	X	X
Multidrop	X	X	X	X
Loop	N.A.	N.A.	C (Only)	C (Only)

*Modem should not respond to break.
**Channels A, B, and C denoted by X.

Memory Capacity/Addressing

On-Board EPROM*

Device	Total K Bytes	Hex Address Range
2716	8	FE000–FFFFF
2732A	16	FC000–FFFFF
2764	32	F8000–FFFFF
27128	64	F0000–FFFFF

With optional iSBC 341 MULTIMODULE EPROM

Device	Total K Bytes	Hex Address Range
2716	16	FC000–FFFFF
2732A	32	F8000–FFFFF
2764	64	F0000–FFFFF
27128	128	E0000–FFFFF

*Four iSBC 88/45 EPROM sockets support JEDEC 24/28-pin standard EPROMs and RAMs (static and iRAM, 3 sockets); iSBC 341 sockets also support EPROMs and RAMs.

Timer Input Frequency—8.00 MHz ±0.1%

Interfaces

IEEE 959 SBX I/O Bus Expansion

Serial RS232C Signals—

CTS	CLEAR TO SEND
DSR	DATA SET READY
DTE TXC	TRANSMIT CLOCK
DTR	DATA TERMINAL READY
FG	FRAME GROUND
RTS	REQUEST TO SEND
RXC	RECEIVE CLOCK
RXD	RECEIVE DATA
SG	SIGNAL GROUND
TXD	TRANSMIT DATA

Serial RS422A/449 Signals—

CS	CLEAR TO SEND
DM	DATA MODE
RC	RECEIVE COMMON
RD	RECEIVE DATA
RS	REQUEST TO SEND
RT	RECEIVE TIMING
SC	SEND COMMON
SD	SEND DATA
SG	SIGNAL GROUND
TR	TERMINAL READY
TT	TERMINAL TIMING

Electrical Characteristics

DC Power Dissipation—28.3 Watts

DC Power Requirements

Configuration	Current Requirements (All Voltages ±5%) +5V	+12V	−12V
Without EPROM(1)	5.1A	20 mA	20 mA
With 8K EPROM (Using 2716)	+0.14A	—	—
With 16K EPROM (Using 2732A)	+0.20A	—	—
With 32K EPROM (Using 2764)	+0.24A	—	—
With 64K EPROM (Using 27128)	+0.24A	—	—

NOTE:
1. AS SHIPPED—no EPROMs in sockets, no iSBC 341 module. Configuration includes terminators for two RS422A/449 and one RS232C channels.

Serial Communication Characteristics

Channel	Device	Supported Interface	Max. Baud Rate
A	8274(1)	RS442A/449 RS232C CCITT V.24	800K SDLC/HDLC 125K Synchronous 50K Asynchronous
B	8274	RS232C CCITT V.24	125K Synchronous(2) 50K Asynchronous
C	8273(3)	RS442A/449 RS232C CCITT V.24	64K SDLC/HDLC(3) 9.6K SELF CLOCKING

NOTES:
1. 8274 supports HDLC/SDLC/SYNC/ASYNC multiprotocol
2. Exceed RS232C/CCITT V.24 rating of 20K baud
3. 8273 supports HDLC/SDLC

BAUD RATE EXAMPLES (Hz)

8254 Timer Divide Count N	Synchronous K Baud	Asynchronous ÷16 K Baud	Asynchronous ÷32 K Baud	Asynchronous ÷64 K Baud
10	800	50.0	25.0	12.5
26	300	19.2	9.6	4.8
31	256	16.1	8.06	4.03
52	154	9.6	4.8	2.4
104	76.8	4.8	2.4	1.2
125	64	4.0	2.0	1.0
143	56	3.5	1.7	0.87
167	48	3.0	1.5	0.75
417	19.2	—	—	—
833	9.6	—	—	—
EQUATION	8,000,000 / N	500K / N	250K / N	125K / N

Line Drivers (Supplied)

Device	Characteristic	Qty	Installed
1488	RS232C	3	1
1489	RS232C	3	1
3486	RS422A	2	2
3487	RS422A	2	2

Reference Manual

143824—iSBC 88/45 Data Communications Processor Board Hardware Reference Manual (not supplied).

ORDERING INFORMATION

Order Code	Description
SBC88/45	iSBC 88/45 Data Communications Processor Board

11

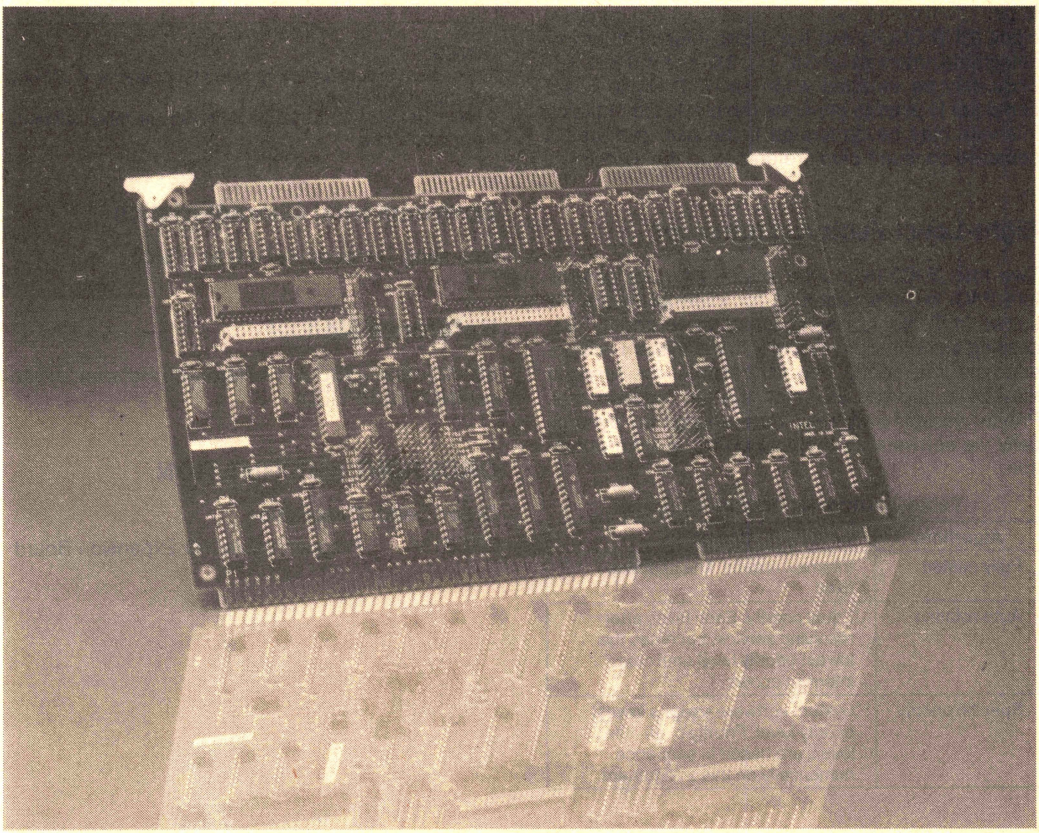
Multibus I Digital and Analog I/O Boards

12

iSBC 519A
PROGRAMMABLE I/O EXPANSION BOARD

- **I/O Expansion via Direct MULTIBUS Interface**
- **72 Programmable I/O Lines with Sockets for Interchangeable Line Drivers and Terminators**
- **Provides Full 16-Bit I/O Addressability**
- **Provides 3 iSBX Multimodule Connectors**
- **Provides 16 Maskable Interrupt Request Lines**
- **Jumper Selectable 0.5, 1.0, 2.0, or 4.0 ms Interval Timer**
- **Provides Eight Maskable Interrupt Request Lines with Priority Encoded and Programmable Interrupt Algorithms**

The iSBC 519A Programmable I/O Expansion Board is a member of Intel's complete line of iSBC memory and I/O expansion boards. The iSBC 519A interfaces directly to any iSBC single board computer via the system bus to expand input and output port capacity. The iSBC 519A provides 72 programmable I/O lines. The system software is used to configure the I/O lines to meet a wide variety of peripheral requirements. The flexibility of the I/O interface is further enhanced by the capability of selecting the appropriate combination of optional line drivers and terminators to provide the required sink current, polarity, and drive/termination characteristics for each application. Address selection is accomplished by using wire-wrap jumpers. The board operates with a single +5V power supply.



280230-1

September 1993
Order Number: 280230-002

FUNCTIONAL DESCRIPTION

The 72 programmable I/O lines on the iSBC 519A are implemented utilizing three Intel 82C55A programmable peripheral interfaces. The system software is used to configure the I/O lines in combinations of undirectional input/output and bidirectional ports. In order to take full advantage of the large number of possible I/O configurations, sockets are provided for interchangeable I/O line drivers and terminators. The 72 programmable I/O lines and signal ground lines are brought out to three 50-pin edge connectors that mate with flat, round, or woven cable.

Interval Timer

Typical I/O read access time is 350 nanoseconds.

Typical I/O read/write cycle time is 450 nanoseconds. The interval timer provided on the iSBC 519A may be used to generate real time clocking in systems requiring the periodic monitoring of I/O functions. The time interval is derived from the constant clock (BUS CCLK) and the timing interval is jumper selectable. Intervals of 0.5, 1.0, 2.0, and 4.0 milliseconds may be selected when an iSBC single board computer is used to generate the clock. Other timing intervals may be generated if the user provides a separate constant clock reference in the system.

Eight-Level Vectored Interrupt

Two Intel 82C59A programmable interrupt controllers (PIC) provide vectoring for interrupt levels. As shown in Table 1, a selection of three priority processing algorithms is available to the system designer so that the manner in which requests are serviced may be configured to match system requirements. Priority assignments may be reconfigured dynamically via software at any time during system operation.

Table 1. Interrupt Priority Options

Algorithm	Operation
Fully nested	Interrupt request line priorities fixed.
Auto-rotating	Equal priority. Each level, after receiving service, becomes the lowest priority level until next interrupt occurs.
Specific priority	System software assigns lowest priority level. Priority of all other levels are based in sequence numerically on this assignment.

iSBX Expansion Bus (IEEE 959) Capabilities

Three iSBX bus connectors are provided on the iSBC 519A board. Up to three single-wide or one double-wide and one single-wide IEEE 959 iSBX board can be added to the iSBC 519A board. A wide variety of expansion options are available.

Physical Characteristics

Width: 12.00 in. (30.48 cm)
Height: 6.75 in. (17.15 cm)
Depth: 0.50 in. (1.27 cm),
1.16 in. (2.95 cm) with iSBX modules
Weight: 14 oz. (397.3 gm)

Electrical Characteristics

Average DC Current

Voltage	Without Termination(1)	With Termination(2)
V_{CC} = +5V ±5%	I_{CC} = 1.5A max	3.5A max

NOTES:
1. Does not include power required for operational I/O drivers and I/O terminators.
2. With 18 220Ω/330Ω input terminators installed, all terminator inputs low.

Environmental Characteristics

Operating Temperature: 0°C to +55°C

Reference Manual

500623-002—iSBC 519A Hardware Reference manual (NOT SUPPLIED)

ORDERING INFORMATION

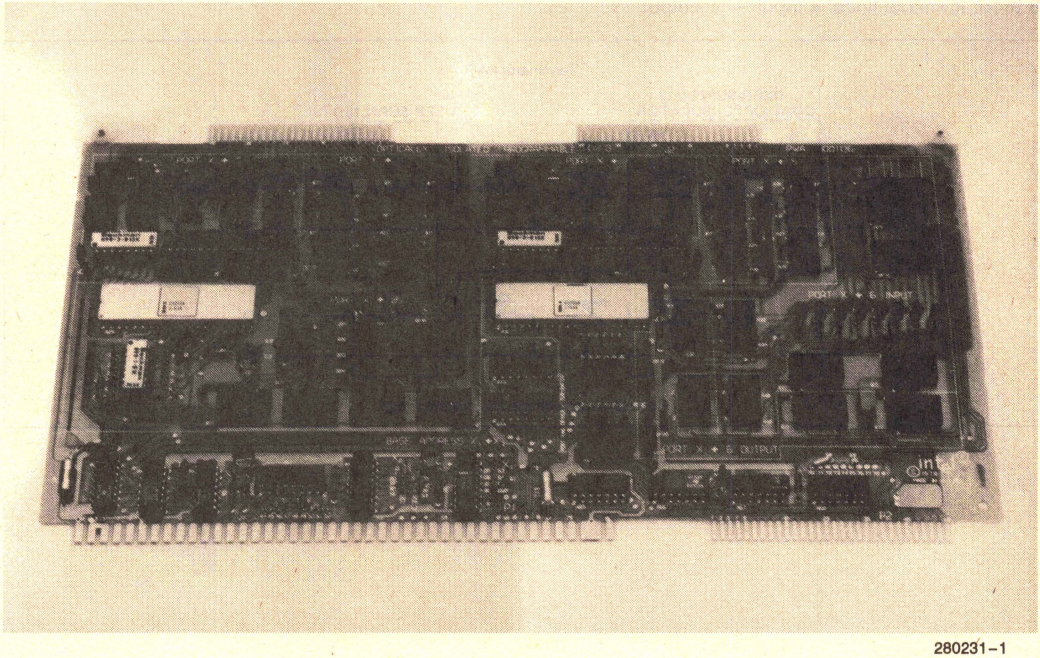
Order Code	Description
SBC519A	Programmable I/O Expansion Board

iSBC 556*
OPTICALLY ISOLATED I/O BOARD

- **Up to 48 Digital Optically Isolated Input/Output Data Lines for MULTIBUS Systems**
- **Choice of**
  - **— 24 Fixed Input Lines**
  - **— 16 Fixed Output Lines**
  - **— 8 Programmable Lines**
- **Provisions for Plug-In, Optically Isolated Receivers, Drivers, and Terminators**
- **Voltage/Current Levels**
  - **— Input up to 48V**
  - **— Output up to 30V, 60 mA**
- **Common Interrupt for up to 8 Sources**
- **+5V Supply Only**

The iSBC 556 Optically Isolated I/O Board provides 48 digital input/output lines with isolation between process application or peripheral device and the system CPU board(s). The iSBC 556 contains two 8255A programmable interface devices, and sockets for user supplied optically isolated drivers, receivers, and input resistor terminators, together with common interrupt logic and interface circuitry for the system bus. Input signals can be single-ended or differential types with user defined input range (resistor terminator and opto-isolated receiver selection), allowing flexibility in design of voltage and threshold levels. The output allows user selection of Opto-Isolated Darlington Pair which can be used as an output driver either as an open collector or current switch.

12



280231-1

*The iSBC 556 is also manufactured under product code piSBC 556 by Intel Puerto Rico, Inc.

August 1993
Order Number: 280231-002

Table 1. I/O Ports Opto-Isolator Receivers, Drivers, and Terminators

Port No. X = I/O Base Address	Type of I/O	Lines (qty)	Resistor Terminator Pac-Rp 16-Pin DIP Bourns 4116R-00 or Equivalent	Dual Opto-Isolator 8-Pin DIP Monsanto MC T66 or Equivalent	Driver 7438 or Equivalent	Pull-Up iSBC 902
X + 0	Input	8	1	4	—	
X + 1	Output	8	—		—	
X + 2	Input/ Control	8	1		—	
X + 4	Input	8	1	4	—	
X + 5	Output	8			—	
X + 6	Input/ Output Control	8	1 if input		2 if input	2 if input
X + 7						

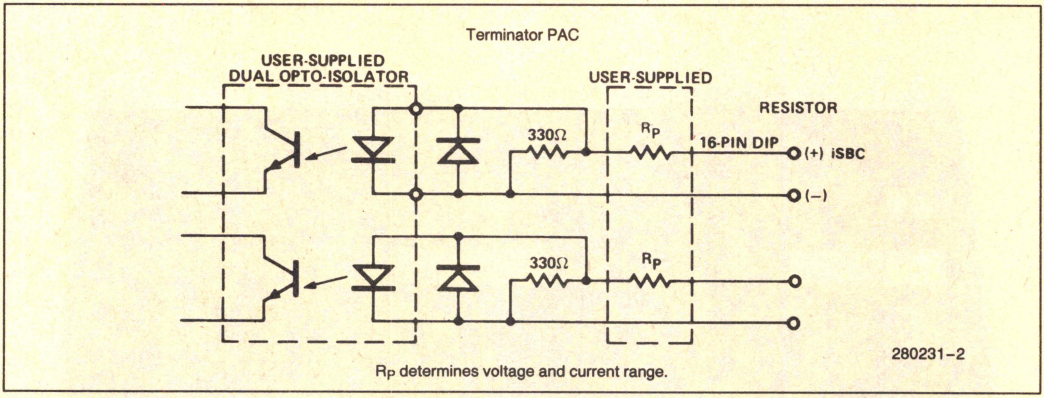
SPECIFICATIONS

Number of Lines

24 input lines
16 output lines
8 programmable lines: 4 input — 4 output

I/O Interface Characteristics

Line-to-Line Isolation: 235V DC or peak AC
Input/Output Isolation: 500V DC or peak AC



RP determines voltage and current range.

I/O Addressing

Port	8255 #1			Control	8255 #2			Control
	A	B	C		A	B	C	
Address	X+0	X+1	X+2	X+3	X+4	X+5	X+6	X+7

Where: base address is from 00H to 1FH (jumper selectable)

Physical Characteristics

Width: 12.00 in. (30.48 cm)
Height: 6.75 in. (17.15 cm)
Depth: 0.50 in. (1.27 cm)
Weight: 12 oz. (397.3 gm)

Electrical Characteristics

Average DC Current

V_{CC} = +5V ±5%, 1.0A without user supplied isolated receiver/driver

I_{CC} = 1.6A max with user supplied isolator receiver/driver

Environmental Characteristics

Temperature: 0°C to 55°C
Relative Humidity: 0% to 90%, non-condensing

Reference Manual

980489-002— iSBC 556 Hardware Reference Manual (Order Separately)

ORDERING INFORMATION

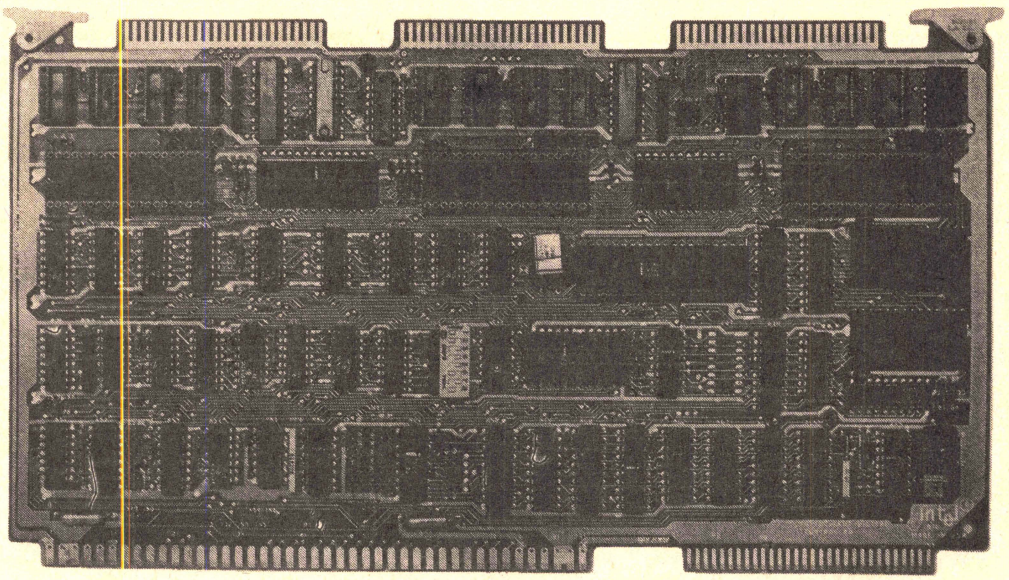
Order Code	Description
SBC556	Optically Isolated I/O Board

12

iSBC 569*
INTELLIGENT DIGITAL CONTROLLER

- Single Board Digital I/O Controller with up to Four Microprocessors to Share the Digital Input/Output Signal Processing
- 3 MHz 8085A Central Control Processor
- Three Sockets for 8041/8741A Universal Peripheral Interface (UPI-41A) for Distributed Digital I/O Processing
- Three Operational Modes
  - — Stand-Alone Digital Controller
  - — MULTIBUS Master
  - — Intelligent Slave (Slave to MULTIBUS Master)
- 2K Bytes of Dual Port Static Read/Write Memory
- Sockets for up to 8K Bytes of Intel 2758, 2716, 2732 Erasable Programmable Read Only Memory
- 48 Programmable Parallel I/O Lines with Sockets for Interchangeable Line Drivers or Terminators
- Three Programmable Counters
- 12 Levels of Programmable Interrupt Control
- Single +5V Supply
- MULTIBUS Standard Control Logic Compatible with Optional iSBC 80 and iSBC 86 CPU, Memory, and I/O Expansion Boards

The Intel iSBC 569 Intelligent Digital Controller is a single board computer (8085A based) with sockets for three 8041A/8741A Universal Peripherals Interface chips (UPI-41A). These devices, which are programmed by the user, may be used to offload the 8085A processor from time consuming tasks such as pulse counting, event sensing and parallel or serial digital I/O data formatting with error checking and handshaking. The iSBC 569 board is a complete digital controller with up to four processors on a single 6.75 inches x 12.00 inches (17.15 cm x 30.48 cm) printed circuit board. The 8085A CPU, system clock, read/write memory, non-volatile memory, priority interrupt logic, programmed timers, MULTIBUS control and interface logic, optional UPI processors and optional line driver and terminators all reside on one board.

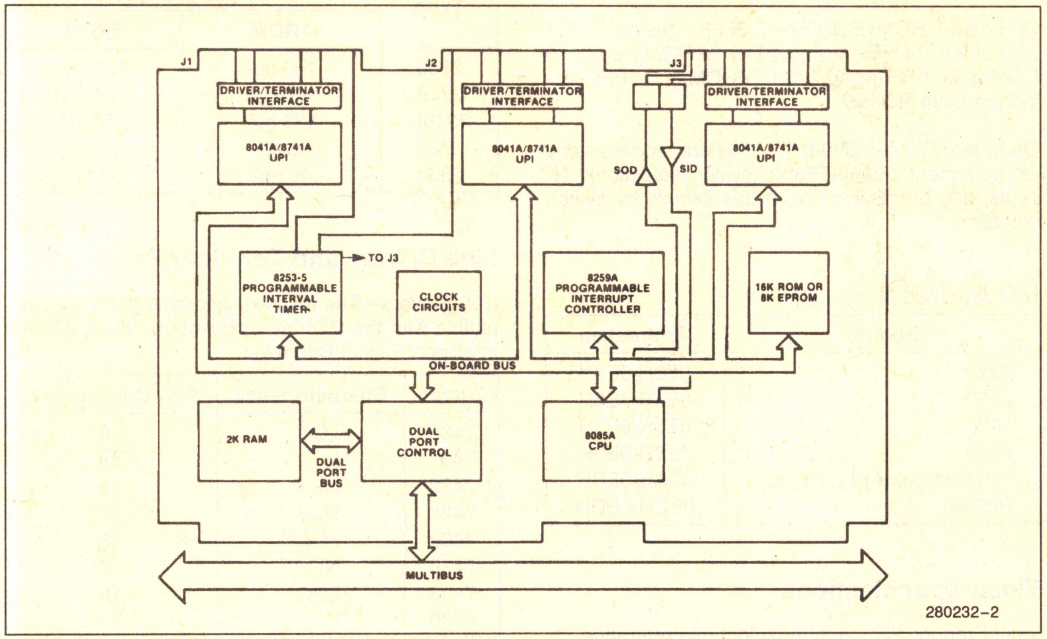


280232-1

*The iSBC 569 is also manufactured under product code piSBC 569 by Intel Puerto Rico, Inc.

September 1993
Order Number: 280232-002

BLOCK DIAGRAM



iSBC 569 Intelligent Digital Controller Block Diagram

SPECIFICATIONS

8085A CPU

Word Size: 8, 16 or 24 bits

Cycle Time: 1.30 μs ±0.1% for fastest executable instruction; i.e., four clock cycles.

Clock Rate: 3.07 MHz ±0.1%

System Access Time

Dual port memory—725 ns

Memory Capacity

On-board ROM/EPROM—2K, 4K, 8K, or 16K bytes of user installed ROM or EPROM.

On-board RAM—2K bytes of static RAM. Fully accessible from on-board 8085A. Separately addressable from system bus.

Off-board expansion—up to 64K bytes of EPROM/ROM or RAM capacity.

I/O Capacity

Parallel-Timers—Three timers, with independent gate input, clock input, and timer output user-accessible. Clock inputs can be strapped to an external source or to an on-board 1.3824 MHz reference. Each timer is connected to a 8259A Programmable Interrupt Controller and may also be optionally connected to UPI processors.

UPI-I/O—Three UPI-41A interfaces, each with two 8-bit I/O ports plus the two UPI Test Inputs. The 8-bit ports are user-configurable (as inputs or outputs) in groups of four.

Serial—1 TTL compatible serial channel utilizing SID and SOD lines of on-board 8085A CPU.

On-Board Addressing

All communications to the UPI-41A processors, to the programmable reset latch, to the timers, and to the interrupt controller are via read and write commands from the on-board 8085A CPU.

Memory Addressing

On-board ROM/EPROM—0-07FF (using 2758 EPROMs); 0-0FFF (using 2716 EPROMs or 2316 ROMs); 0-1FFF (using 2732 EPROMs); 0-3FFF (using the 2364 ROMs)

On-board RAM—8000-87FF System access—any 2K increment 00000-FF800 (switch selection); 1K bytes may be disabled from bus access by switch selection.

I/O Addressing

Source	Addresses
8253	0E0H-0E3H
UPI0	0E4H-0E5H
UPI1	0E6H-0E7H
UPI2	0E8H-0E9H
PROGRAMMABLE RESET	0EAH-0EBH
8259A	0ECH-0EDH

Timer Specifications

Input Frequencies—jumper selectable reference

Internal: 1.3824 MHz ±0.1% (0.723 μs, nominal)
External: User supplied (2 MHz maximum)

Output Frequencies (at 1.3824 MHz)

Function	Min[1]	Max[1]
Real-time interrupt interval	1.45 μsec	47.4 msec
Rate Generator (frequency)	21.09 Hz	691.2 KHz
1. Single 16-bit binary count		

Physical Characteristics

Width: 30.48 cm (12.00 inches)
Depth: 17.15 cm (6.75 inches)
Thickness: 1.27 cm (0.50 inch)
Weight: 3.97 gm (14 ounces)

Electrical Characteristics

DC Power Requirements—+5V @ 2.58A with no optional devices installed. For each 8741A add 135 mA. For each 220/330 resistor network, add 60 mA. Add the following for each EPROM/ROM installed.

Type	+5.0V Current Requirement	
	1ROM	2ROM
2758	100 mA	125 mA
2716	100 mA	125 mA
2316E	120 mA	240 mA
2732	40 mA	55 mA
2364	40 mA	55 mA

Line Drivers and Terminators

I/O /Drivers—The following line drivers are all compatible with the I/O driver sockets on the iSBC 569 Intelligent Digital Controller.

Driver	Characteristics	Sink Current (mA)
7438	I,OC	48
7437	I	48
7432	NI	16
7426	I,OC	16
7409	NI,OC	16
7408	NI	16
7403	I,OC	16
7400	I	16

NOTE:
I = inverting; NI = non-inverting; OC = open collector.

I/O Terminators—220Ω/330Ω divider or 1 kΩ pull-up (DIP) - user supplied

Environmental Characteristics

Operating Temperature: 0° C to 55° C (32° F to 131 °F)
Relative Humidity: To 90% without condensation

Reference Manual

980845-002— iSBC 569 Intelligent Digital Controller Board Hardware Reference Manual (NOT SUPPLIED)

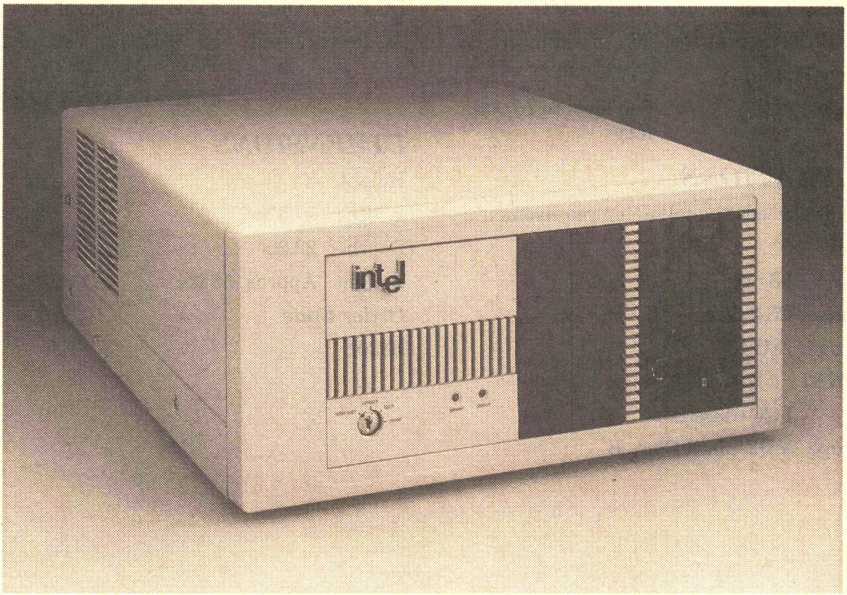
ORDERING INFORMATION

Order Code	Description
SBC569	Intelligent Digital Controller

Multibus I System Packaging and Power Supplies

13

SYSTEM 320 CHASSIS FOR MULTIBUS I



280502-1

13

RUGGED AND DURABLE MULTIBUS I CHASSIS WITH EXCEPTIONAL EMI AND RFI SHIELDING

The B320E is an excellent Multibus I chassis for OEMs building integrated Multibus I systems. This sturdy chassis proved its worth in thousands of performance critical applications from Wall Street to the factory floor. The seven slot capacity makes the B320E chassis ideal for a wide range of applications ranging from development systems to mission critical data acquisition applications and process control functions.

The B320E is constructed to withstand harsh environments. Operating temperatures from 10°C to 40°C, wetbulb temperature 26°C maximum, relative humidity 85% at 40°C, and altitudes from sea level to 10,000 feet are possible operating conditions. The chassis layout allows for good cooling, drive capacity, and serviceability.

SYSTEM 320 FEATURES

- Seven Slots, Multibus I IEEE 796 Specification
- 435 Watt Power Supply with switchable input for ~110 or ~220 VAC
- FCC Class B and VDE Limit Class B
- Four ½ height (two full height) 5 ¼ drives bays with front panel access
- One full height 5 ¼ internal drive bays
- Control panel with key switch and indicator lights

*MSNET is a trademark of Microsoft Corporation.
**VAX/VMS is a trademark of Digital Equipment Corporation.
*The System 320 is also manufactured under product code pSYS320 by Intel Puerto Rico, Inc.

August 1993
Order Number: 280502-003

intel

SPECIFICATIONS

ENVIRONMENT

Operating Temperature	10°C to 40°C
Wet Bulb Temperature	26°C maximum
Relative Humidity	85% at 40°C
Altitude	Sea level to 10,000 feet

REGULATIONS

Meets or exceeds the following requirements

Safety

US	UL 478
Canada	CSA C22.2
Europe	IEC 435

EMI/RFI

US	FCC Class B
Europe	VDE Limit Class B

ELECTRICAL

DC Power Output	435W Maximum
AC Power Input	88 VAC–132 VAC or 176 VAC–264 VAC
	47 Hz–63 Hz, single phase

DIMENSIONS

Height	8"
Width	17.5"
Depth	22.25"
Weight	Approx. 55 lbs.

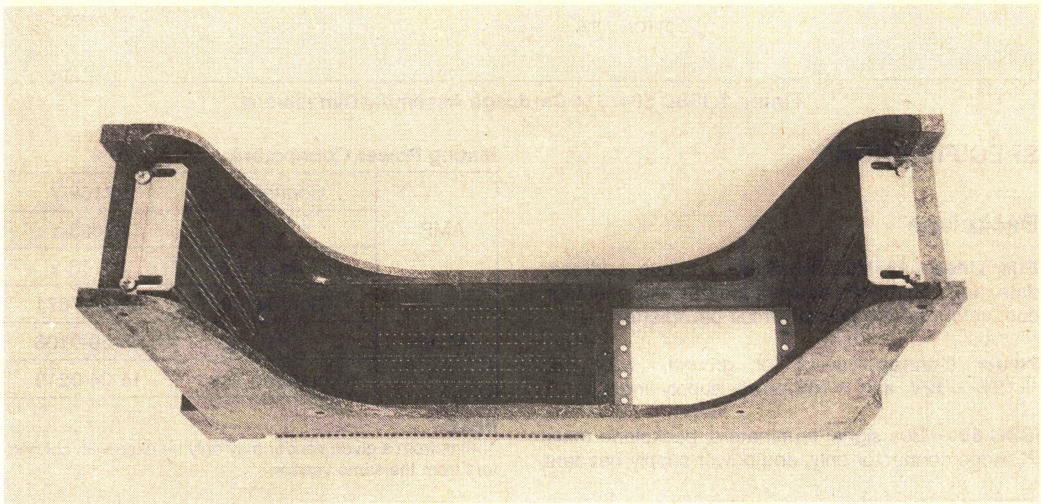
Order Code

B320E

iSBC 604/614
MODULAR CARDCAGE ASSEMBLIES

- **Interconnects and Houses up to Four MULTIBUS Boards per Cardcage**
- **Connectors Allow Interconnection of up to Four Cardcage Assemblies for 16 Board Systems**
- **Strong Cardcage Structure Helps Protect Installed Boards from Warping and Physical Damage**
- **Cardcage Mounting Holes Facilitate Interconnection of Units**
- **Compatible with 3.5-Inch RETMA Rack Mount Increments**
- **Interleaved Grounds on Backplane Minimize Noise and Crosstalk**
- **Up to 3 CPU Boards per System for Multiprocessing Applications**

The iSBC 604 and iSBC 614 Modular Cardcage Assemblies units provide low-cost, off-the-shelf housing for OEM products using two or more MULTIBUS boards. Each unit inerconnects and houses up to four boards. The base unit, the iSBC 604 Cardcage Assembly, contains a male backplane PC edge connector and bus signal termination circuits, plus power supply connectors. It is suitable for applications requiring a single unit, or may be interconnected with up to three iSBC 614 cardcage assemblies for a four cardcage (16 board) system. The iSBC 614 contains both male and female backplane connectors, and may be interconnected with iSBC 604/614 units. Both units are identical, with the exception of the bus signal terminator feature. A single unit may be packaged in a 3.5 inch RETMA rack enclosure, and two interconnected units may be packaged in a 7 inch enclosure. The units are mountable in any of three planes.



280205-1

August 1993
Order Number: 280205-001

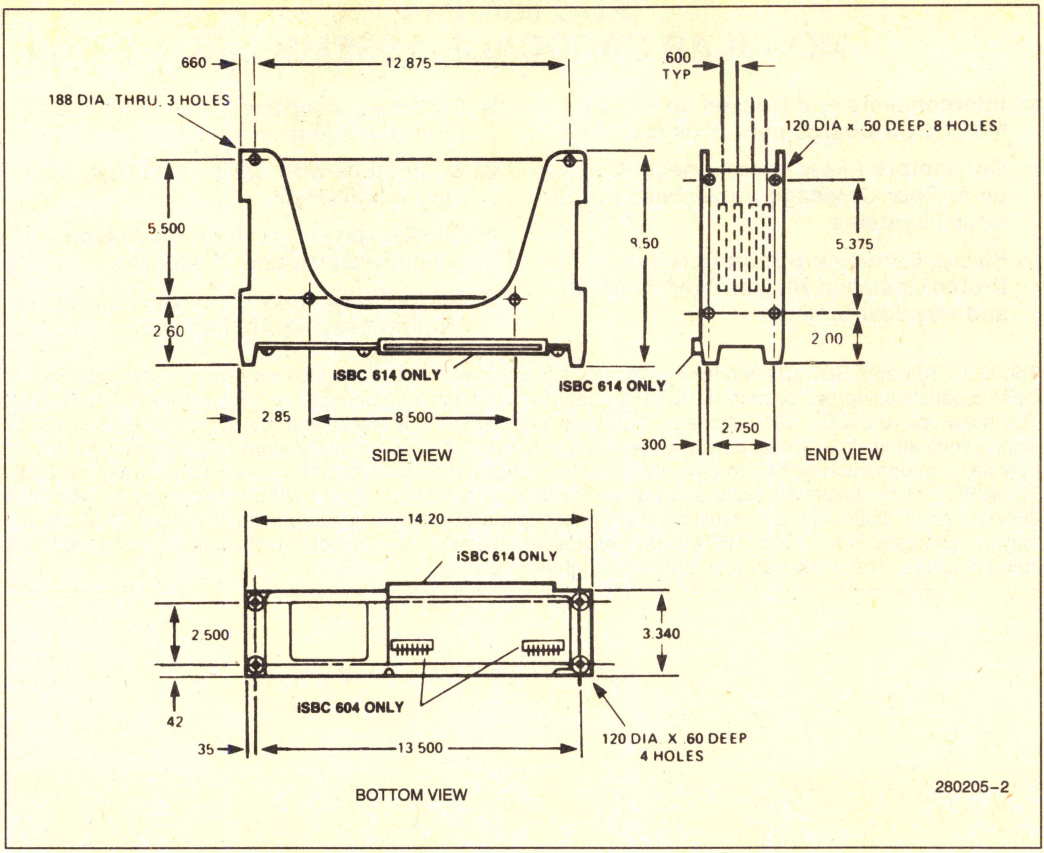


Figure 1. iSBC 604/614 Cardcage Assembly Dimensions

SPECIFICATIONS

Backplane

Bus Lines—All MULTIBUS system bus address, data, and command bus lines are bussed to all four connectors on the printed circuit backplane

Power Connectors—G for ground, +5, −5, +12V, −12V, and −10V power supply lines

iSBC 604—Bus signal terminators, backplane male PC edge connector only, and power supply headers

iSBC 614—Backplane male and female connectors and power supply headers

Mating Power Connectors

AMP	Connector	87159-7
	Pin	87023-1
	Polarizing Key	87116-2
Molex	Connector	09-50-7071
	Pin	08-50-0106
	Polarizing Key	15-04-0219

NOTE:
1. Pins from a given vendor may only be used with connectors from the same vendor.

ORDERING INFORMATION

Part Number	Description
SBC 604	Modular Cardcage Assembly (Base Unit)

Bus Arbitration: Serial; up to 3 CPU masters

Equipment Supplied: iSBC 604 or iSBC 614 Cardcage Schematic

Physical Dimensions

Height: 8.5 in. (21.59 cm)

Width: 14.2 in. (36.07 cm)

Depth: 3.34 in. (8.48 cm)

Weight: 35 oz. (992.23 gm)

Card Slot Spacing: 0.6 in.

Environmental Characteristics

Operating Temperature: 0°C to 55°C

Reference Manual

9800708—iSBC 604/614 Cardcage Hardware Reference Manual (ORDER SEPARATELY)

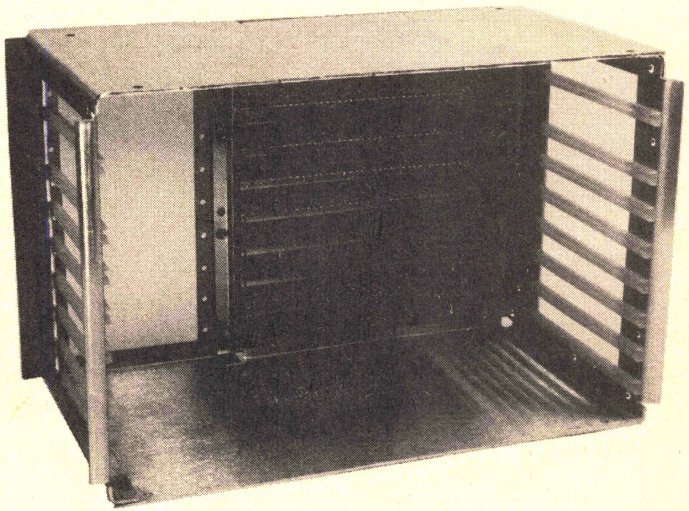
Part Number	Description
SBC 614	Modular Cardcage Assembly (Expansion Unit)

iSBC 608/618 CARDCAGES

- **Houses Eight MULTIBUS iSBC Boards in an Aluminum Package**
- **Board-to-Board Clearance for iSBC MULTIMODULE Boards on All Slots**
- **Board-to-Board Clearance for iSBX MULTIMODULE Boards on Two Slots**
- **Parallel Priority Circuitry for up to Eight Multimaster iSBC Boards**
- **Enhanced Bus Noise Immunity for High Speed Systems**
- **Plug on iSBC 618 Unit for up to Sixteen Board Systems**
- **NEMA-Type Backwall or 19-Inch Rack Mount Hardware Included**
- **Signal Line Termination Circuitry on iSBC 608 Cardcage**

Intel's iSBC 608/618 Cardcages are matched to the latest generation of iSBC/iSBX boards which mount in the MULTIBUS system bus. These products provide several features which make them the industry's leading price/performance cardcage product. MULTIMODULE board clearance, parallel priority circuitry, enhanced backplane noise immunity, and precision fit card guides are a few of the distinctions which make this the industry's better product.

The iSBC 608 Cardcage is the base unit, housing up to eight iSBC boards and their MULTIMODULE boards. Additionally, this base unit includes mounting hardware and fan mounting bracketry. The iSBC 618 is the expansion unit, providing eight additional iSBC board slots to the iSBC 608 Cardcage for a total of sixteen board slots which can be NEMA-type backwall or 19-inch rack mounted. This is accomplished with the mounting hardware of the iSBC 608 Cardcage. The iSBC 618 expansion unit also includes fan mounting bracketry.



210373-1

September 1993
Order Number: 210373-001

FUNCTIONAL DESCRIPTION

Mechanical Aspects

The iSBC 608/618 Cardcages provide housing and a MULTIBUS system bus for up to sixteen single board computers and their MULTIMODULE boards. The iSBC 608 unit and iSBC 618 unit offer board-to-board clearance (0.8 inches or greater) on all eight slots for iSBC MULTIMODULE boards. Two slots provide clearance (1.2 inches or greater) for iSBX MULTIMODULE boards as shown in Figure 1. Each cardcage includes precision fitted nylon cardguides for secure board fit and accurate MULTIBUS board pin alignment. Fan mounting bracketry is also included with each cardcage. This bracketry allows the mounting of several industry standard fans. The iSBC 608 Cardcage base unit includes aluminum mounting hardware for NEMA-type backwall mounting, or anchoring a sixteen slot iSBC 608/618 combination in a standard 19-inch rack.

Electrical Aspects

The iSBC 608/618 Cardcages implement a parallel priority resolution scheme by using plug-in jumper connections. There are six different priority schemes allowed, each requiring a different jumper configuration. In systems where an iSBC 618 Cardcage is attached to the base unit, the base unit will have lower priority overall. That is, master boards in the iSBC 608 base unit bay gain control of the MULTIBUS lines only when no boards in the iSBC 618 expansion unit are asserting the bus request (BREQ/) signal.

Noise-minimizing ground traces are strategically interleaved between signal and address lines on these backplanes. This provides the enhanced noise immunity and minimized signal-to-signal coupling which is important in high speed, high board count microcomputer systems.

The iSBC 608/618 Cardcages provide power connector lug bolts for +5 VDC and ground. The lug bolts, compared to other power connection methods, help transfer higher amounts of current. Other voltages (±12 VDC, −5 VDC) are connected via a mating power connector plug as shown in Figure 2.

13

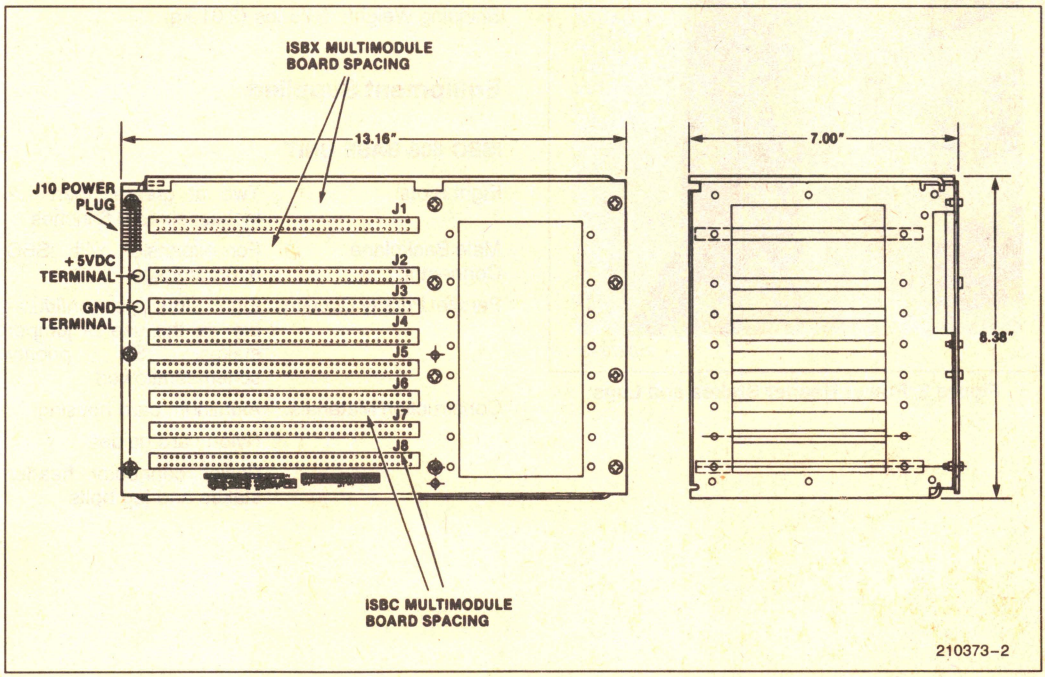


Figure 1. iSBC 608/618 Cardcages Dimensions

SPECIFICATIONS

Bus Lines

All MULTIBUS (IEEE 796) system bus address and command lines are bussed to each of the eight MULTIBUS connectors on the backplane. Ground traces are interleaved among these signal lines and bussed to the backplane edge connector for interconnection of the iSBC 608 and iSBC 618 backplane.

Power Connectors

Ground (0V), +5V, −5V, +12V, −12V power supply header stakes and power lug bolts are provided on the iSBC 608/618 Cardcages as shown in Figure 2.

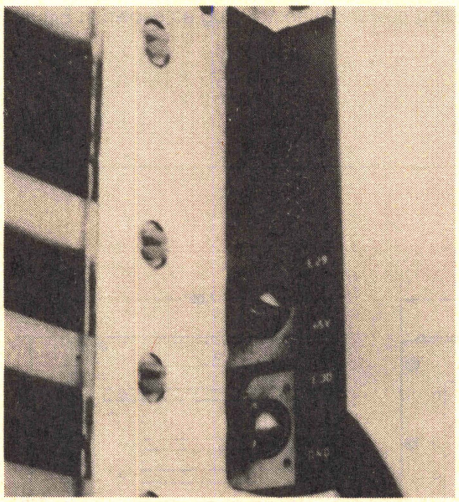


Figure 2. Power Header Stakes and Lugs

Environmental Characteristics

Operating Temperature:	0°C to 55°C
Storage Temperature:	−40°C to +85°C
Humidity:	50% to 95% non-condensing at 25°C to 40°C.
Vibration and Shock:	2G max. through 50 Hz

Physical Characteristics

SLOT-TO-SLOT DIMENSIONS (See Figure 1)

Top-J1:	1.200 in. (to center)
J1-J2:	1.300 in. (center to center)
J8-Bottom:	0.700 in. (to center)
All Others:	0.800 (center to center)

Physical Dimensions

Height:	8.38 in. (21.29 cm)
Length:	13.16 in. (33.43 cm)
Width:	7.50 in. (19.05 cm)
Weight:	3.50 lbs (1.59 kg)
Shipping Weight:	5.75 lbs (2.61 kg)

Equipment Supplied

iSBC 608 BASE UNIT

Eight Slots:	Two at greater than 1.2 inches; six at 0.8 inches
Male Backplane Connector:	For expansion with iSBC 618 cardcage
Parallel Priority Circuitry:	Eight slots are configurable via the use of jumper stakes. Six priority schemes allowed
Construction Materials:	Aluminum card housing
	Nylon card guides
	Power connector header stakes and lug bolts

Accessories

iSBC 618 EXPANSION UNIT

Eight-Slots:	Two at greater than 1.2 inches; six at 0.8 inches
Female Backplane Connector:	For expansion to iSBC 608 base unit
Parallel Priority Circuitry:	Eight slots are configurable via the use of jumper stakes. Six priority schemes allowed.
Construction Materials:	Aluminum card housing
	Nylon card guides
	Power connector header stakes and lug bolts
	Fan Mounting Hardware
	Schematic

User-Supplied Equipment

MATING POWER CONNECTORS

Vendor	Part Number
3M	3399-6026
Ansley	609-2600M
Berg	65485-009

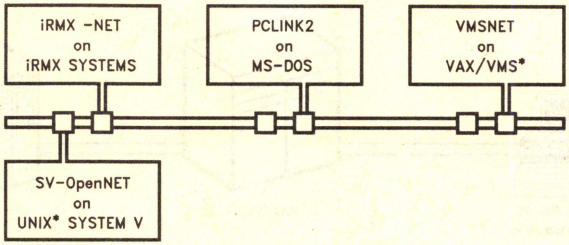
MOUNTABLE FANS

Vendor	Part Number
Rotron	SU2A1-028267
Torin	TA300-A30473-10
Pamotor	8506D

Local Area Network Boards and Software

14

OpenNET LOCAL AREA NETWORK FAMILY



280669-1

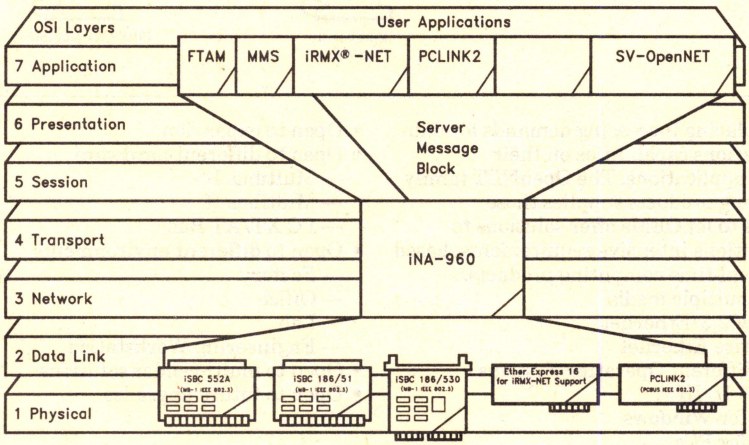
OpenNET: THE COMPLETE OPEN NETWORK SOLUTION

The OpenNET family provides the OEM with complete *Open Network* solutions for an enterprise-wide, multi-vendor network based on international standards.

FEATURES

- Interoperability between the factory, office, and engineering environments
- Complete hardware and software network solutions
- Ongoing customer support through extensive training and application development

GUIDE TO THE OpenNET PRODUCTS



280669-2

14

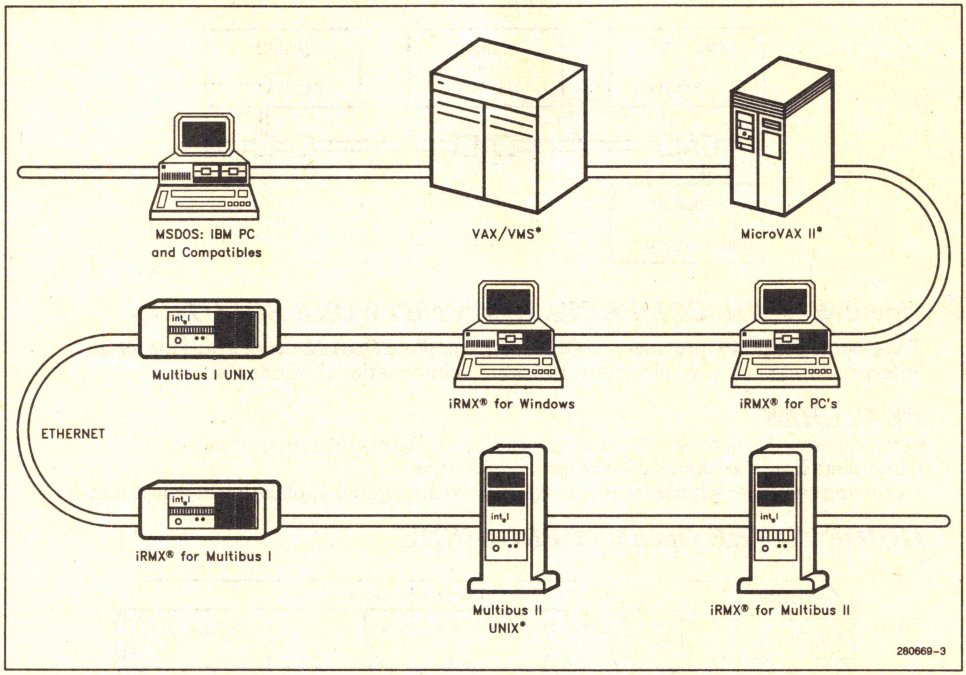
*UNIX is a registered trademark of AT&T
MS-DOS, XENIX are trademarks of Microsoft Corporation
DECnet, VAX/VMS, MicroVAX, UNIBUS, Q-bus are trademarks of Digital Equipment Corporation

September 1993
Order Number: 280669-003

intel

OpenNET OVERVIEW

OpenNET MEANS OPEN NETWORKS



Users are placing increasing demands for data communications capabilities on their computing applications. The OpenNET family of networking products supplies those capabilities to let OEMs offer solutions to communications-intensive requirements, based on Intel's real-time computing products.

- Open to multiple media
  - — IEEE 802.3/Ethernet
  - — Thin-wire Ethernet
- Open to different Operating Systems
  - — iRMX® I//III
  - — iRMX for Windows
  - — iRMX for PCs
  - — MS-DOS*
  - — PC-DOS
  - — UNIX SYSTEM V*
  - — VAX/VMS*
- Open to expansion
- Open to different hardware
  - — Multibus I
  - — Multibus II
  - — PC XT/AT Bus
- Open to different environments
  - — Factory
  - — Office
  - — Lab
  - — Engineering Workstation
- Open to multi-vendor solutions
- Open to future upgrades

OpenNET OVERVIEW

OpenNET ARCHITECTURE

Intel's OpenNET communications architecture applies networking standards to offer an open network. Open connectivity lets Multibus-based systems talk to systems such as IBM PCs and VAX minicomputers.

OpenNET DELIVERS INTERNATIONAL STANDARDS

The OpenNET products use ISO and CCITT Data Communications Standards for the Physical, Data Link, Network, and Transport layers of the Open System Interconnection (OSI) model. The Session, Presentation, and Application layers use the Server Message Block (SMB) protocol promulgated by Intel, Microsoft, and IBM. The SMB protocol is used by Microsoft and IBM in their PC networks.

SMB: TRUE TRANSPARENT REMOTE FILE ACCESS

The OpenNET Server Message Block protocol allows applications to access remote files as if they were local. This consistent view of the file system throughout the network allows distributed processing of existing applications without change. The SMB protocol protects the user's data with file security established by either the consumer or server system.

OpenNET WILL FOLLOW THE STANDARDS, TRANSPARENTLY

As the OSI standards evolve, Intel will conform to them. The user's application will be protected from the changing protocols because the OpenNET Interface ("\\" on DOS or "//" on UNIX) will NOT change. This will allow applications to move to the OSI standards with little or no modification. The OpenNET products will allow the user to run all current applications that use the OpenNET Interface on either the SMB or the OSI protocols.

OpenNET SUPPORT FOR DISKLESS WORKSTATION

For certain real-time applications, it is desirable that a networked system not have a local mass storage device such as a hard disk. One example is harsh environments such as factory floor process control, where rotating media can cause system reliability problems. Another example is financial workstations, where stock and market information must be updated in real-time on a centralized file server and accessed by diskless trader workstations.

The Boot Server system responds to network boot requests from diskless Boot Consumer systems and downloads both the appropriate operating system and communications software over the network to boot the diskless system. With the diskless system booted, the File Server system provides the diskless system with network access to mass storage devices. The Boot Server and File Server can reside in separate systems or in the same system on the network.

OpenNET VIRTUAL TERMINAL CAPABILITY

The OpenNET product family supports a Virtual Terminal (VT) capability. This feature allows a terminal or PC user to "login", over the network, as an interactive user under the remote node's operating system. VT is purchased separately as an optional application program.

All OpenNET PRODUCTS BACKED BY INTEL CUSTOMER SUPPORT

All Intel software products include Intel's Software Support for a 90 day period immediately following the licensing and receipt of the product.

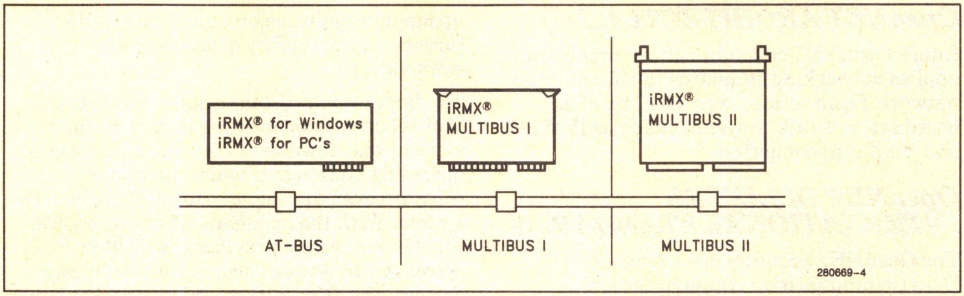
STANDARD SOFTWARE SUPPORT INCLUDES

- Product updates
- Subscription Service and technical product information distributed via:
  - — Monthly issue of *Comments* newsletter
  - — Quarterly Troubleshooting Guides
  - — Software Problem Report (SPR) Service
- Technical Information Phone Service (iTIPS) toll-free hotline
- Membership in Insite User Program Library

ADDITIONAL SERVICES AVAILABLE INCLUDE

- Consulting services on a long or short-term basis (Systems Engineering Support)
- Worldwide training workshops on a wide variety of Intel products
- A full range of hardware maintenance services for end users or OEM/VAR customers through Computervision and Phoenix Repair Center

iRMX-NET OpenNET NETWORKING SOFTWARE



COMPLETE OpenNET SOLUTION FOR REAL-TIME SYSTEMS

Real-Time computer systems require a real-time operating system. The family of iRMX operating systems from Intel is the world's most popular operating system software for real-time systems.

Many real-time applications require network communication. Intel's iRMX-NET Release 3.1 delivers a rich set of networking capabilities and a full range of iRMX platform support:

- Transparent Network File Access
- Transport and Distributed Name Server Software with Programmatic Access
- iRMX for Windows, iRMX for PCs, Multibus I, and Multibus II Connections
- Remote Boot for Diskless Systems

Networked iRMX systems serve in a wide range of real-time application areas including data acquisition, factory automation, financial workstations, military, medical instrumentation, simulation and process control.

TRANSPARENT NETWORK FILE ACCESS

iRMX-NET implements the NFA protocol to provide transparent file access capabilities among iRMX, DOS, VAX/VMS, and UNIX systems on the OpenNET network. Remote files are accessed as if they resided on the local iRMX system. iRMX-NET can be configured as a network file consumer, file server, or both, depending on the application's requirements.

The iRMX operating system provides a rich set of human interface commands and system calls for accessing local files. With the addition of iRMX-NET, these commands and system calls are transparently extended to remote access as well. Transparency means that applications using the iRMX Human Interface commands or BIOS system calls do not need to know whether the files they access reside locally or on some remote system.

iRMX-NET OpenNET NETWORKING SOFTWARE

OSI TRANSPORT AND DISTRIBUTED NAME SERVER WITH PROGRAMMMATIC INTERFACE

The iRMX-NET R3.1 product includes iNA 960 R3 OSI Transport and Network software preconfigured for a variety of Intel Network Interface Adapters.

iRMX-NET R3.1 also includes the iRMX-NET Distributed Name Server software. The Distributed Name Server software maintains and provides access to a network directory database. The database is distributed across the network with each system maintaining its own logical piece of the directory. The Distributed Name Server software provides a full set of network directory services and is used to perform such tasks as logical name to network address mapping for establishing network connections between systems.

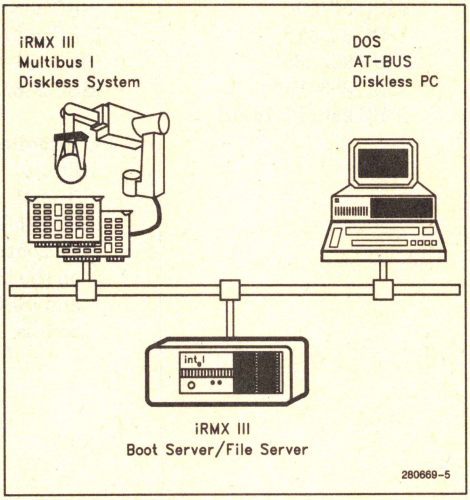
The combination of transparent network file access with iRMX commands and system calls, plus direct programmatic access to the iNA 960 Transport and iRMX-NET Distributed Name Server software gives the programmer a powerful set of capabilities for developing real-time network applications.

iRMX CONNECTIONS

iRMX-NET R3.1 provides networking support for the full range of Intel real-time Systems, from the low-cost IBM compatible i386 PC AT, through the Multibus I to the high-end multiprocessing Multibus II. iRMX-NET R3.1 also supports iRMX board-level designs built around Intel's family of host CPU boards and Network Interface Adapters. Consistent operating system and networking software interfaces provide for easy development of network applications that span the various iRMX platforms.

REMOTE BOOT FOR DISKLESS SYSTEMS

iRMX-NET R3.1 supports networked diskless systems by providing network Boot Consumer, Boot Server and File Server capabilities.



PRODUCT CODES

RMXINETSW	iRMX-NET Networking Software for the iRMX I operating system.
RMXIINETSW	iRMX-NET Networking Software for the iRMX II operating system.

intel

iRMX-NET OpenNET NETWORKING SOFTWARE

REAL-TIME BOARD AND SYSTEM LEVEL SUPPORT

iRMX® I	iRMX® III	
Multibus I	**Multibus I**	**Multibus II**
CPU Boards iSBC 86/30 iSBC 86/35A iSBC 86/C38 iSBC 286/10A iSBC 286/12, 14, 16	**CPU Boards** iSBC 386/12(S) iSBC 486/12(S) iSBC 386/2X iSBC 386/3X **Controller Boards** iSBC 214 iSBC 221 iSBC 215G SCSI controller on the iSBC 386/12S and iSBC 486/12S boards	**CPU Boards** iSBC 386/120 iSBC 386/133 iSBC 386/258 iSBC 386/258D iSBC 486/125 iSBC 486/150 iSBC 486/133SE iSBC 486/166SE MIX 486/SX33 MIX 486/DX33 MIX 486/DX66 MIX 386/020A MIX 486/020A MIX 486/DX66 MIX 486/DX33 MIX 486/SX25 **I/O Server Boards** iSBC 386/258 iSBC 386/258D iSBC 486/133SE iSBC 486/166SE

iRMX for Windows	iRMX for PCs
• Intel OEM PC platforms with a PCL2(A) board • Multibus platforms —iSBC 386/SX (Multibus I) —iSBC 486/SX25/486/DX33/486/DX66 (Multibus II)	iRMX for PCs runs on systems containing Intel386™ or Intel486™ processors. The following systems are supported: • Intel OEM PC platforms • IBM-compatible and Compac-compatible PC platforms • Intel Multibus products: —iSBC 386/SX (Multibus I) —iSBC 486/SX25/486/DX33/486/DX66 (Multibus II)

PCLINK2, NetBIOS, MS-NET ACCESS FOR THE PC

COMPLETE OpenNET SOLUTION FOR THE PC

Users of IBM PC AT, PC XT and other compatible computers can access Intel's OpenNET networking family through the OpenNET PCL2 hardware and software products. The hardware connection is provided by an 80186/82586-based intelligent expansion board, the PCL2 Network Interface Adapter (PCL2NIA and PCL2ANIA). The software package incorporates: MS-NET for transparent file access under DOS, iNA 961, NetBIOS interface, dynamic name resolution and user-friendly installation software.

The NetBIOS interface provides the flexibility to use the PCL2NIA and PCL2ANIA with commercially available NetBIOS compatible applications, such as IBM's PC-LAN program. Optionally, MS-NET networking software is available for the upper layers.

TRANSPARENT NETWORKING FILE ACCESS

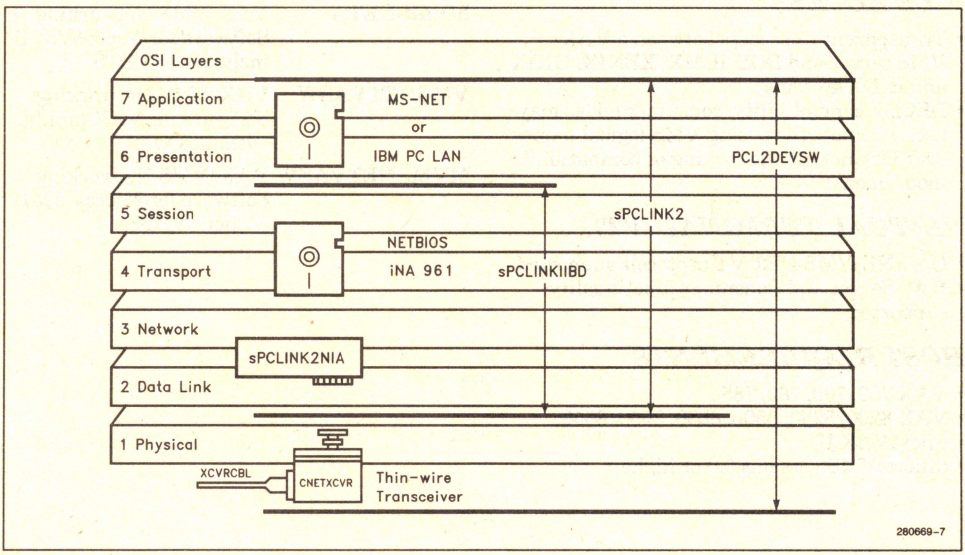
OpenNET/PCL2 gives users the freedom to network PCs as consumer workstations or as file servers. PCLINK2 with MS-NET implements the SMB Protocol for easy access to files on other operating systems, such as and UNIX, iRMX, or VAX/VMS.

REMOTE BOOT FOR DISKLESS SYSTEM

Diskless workstation support for the PC is provided by on-board firmware, an iRMX-NET Boot Server and any OpenNET File Server on the network.

PRODUCT CODES

PCL2NIA	PCL2 Network Interface Adapter with 256 Kbyte of memory, hardware only
PCL2ANIA	PCL2 Network Interface Adapter with 512 Kbyte of memory
PCL2DEVSW	NetBIOS Developer's Kit with: 2-sPCLINK2TWKITs, NetBIOS and Request Block Developer's Software with iNA 961 for PCLINK2 and PCLINK2A
MSNET	OpenNET for MS-DOS



VAX/VMS* OpenNET NETWORKING SOFTWARE

COMPLETE OpenNET SOLUTION FOR THE VAX

VAX/VMS Networking software (VMSNET) is designed to serve on top of Digital Equipment's VAX OSI Transport Services (VOTS) and provides the OpenNET connection for a VAX* or MicroVAX II* system to iRMX, DOS, and UNIX System V.

VMSNET enables a MicroVAX or VAX system to act as an OpenNET file server system allowing any OpenNET consumer node transparent file access to files on the MicroVAX or VAX.

The use of VOTS in the product allows the OpenNET software to run on any VAX running VOTS 5.0 and having a DEC supplied LAN controller.

VMSNET software performs the OpenNET functions via an implementation of the Network File Access (NFA) file server protocols, VMS consumer bi-directional file transfer utilities, and Intel's iNA 960 transport layer software running on the supplied intelligent LAN controllers.

A set of network management utilities provide (Micro)VAX users with information and statistics about VMSNET activities.

FILE ACCESS

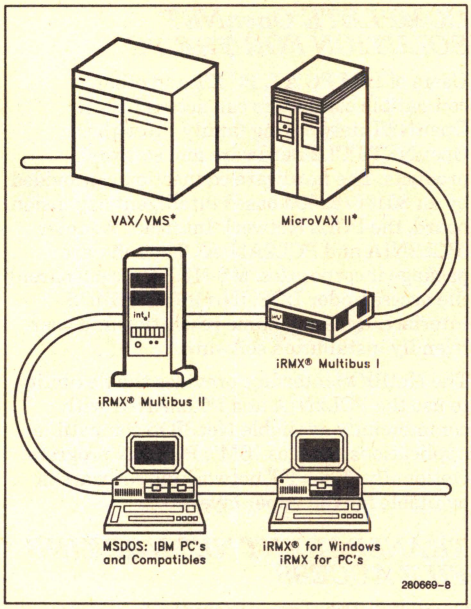
- Transparent file access between a VAX/VMS server and DOS, iRMX, XENIX, UNIX and iNDX systems
- DECnet compatibility: consumer nodes may access remote files using VMS logical names over DECnet (no file locking or compatibility mode opens)

VIRTUAL TERMINAL (VT)

- OpenNET/MS-NET VT protocols supported
- Both Server and Consumer functionality supported

HOST REQUIREMENTS

- VAX 750, 780, 782, 785
- VAX 8200, 8250, 8500, 8530, 8600, 8650
- MicroVAX II
- (Micro)VMS versions 5.0 or higher



PRODUCT CODES

VMSNETVA	VAX/VMS Networking Software for VAX family inclusive of VOTS
MVMSNETA	VAX/VMS Networking Software for MicroVAX II inclusive of VOTS
VMSNETVASW	VAX/VMS Networking Software for VAX family without VOTS
MVMSNETVASW	VAX/VMS Networking Software for MicroVAX II without VOTS

intel

INTEL SYSTEM V OpenNET NETWORKING SOFTWARE

COMPLETE OpenNET SOLUTION FOR UNIX SYSTEM V

SV-OpenNET connects Intel SYSTEM V/386 systems with all the OpenNET nodes. SV-OpenNET is available for Multibus I and Multibus II. The product includes a complete solution: communications board, Mail, VT, print spooling, nameserver interface library (NSI), and network management.

SV-OpenNET allows application interfacing through the UNIX TLI library. Applications may also access SV-OpenNET via the higher-level NSI library. SV-OpenNET can also coexist with the UNIX network, RFS.

FEATURES

Server Message Block (SMB) Transparent File Access

- Transparent file access between DOS, iRMX, and UNIX.
- Both Server and Consumer functionality supported
- Remote Batch Execution (RBE) through "rexec"

MAIL

- Supports MMDF

VIRTUAL TERMINAL (VT)

- OpenNET/MS-NET VT protocols supported
- Both Server and Consumer functionality supported

PRINT SPOOLING

- Interface through "rprint"
- Supports Core printer spooling protocol

UNIX STANDARD INTERFACE

- Interface via AT&T supplied TLI (Streams) library, allowing all TLI applications to interoperate with SV-OpenNET
- SV-OpenNET provides a library, NSI, for high-level Virtual Circuit (VC) creation and name to address translation. The NSI then communicates directly with the UNIX TLI

HOST REQUIREMENTS

Intel SYSTEM V.3.1 UNIX Operating System on Multibus I or Multibus II

PRODUCT CODES

SVNET SV-OpenNET for Multibus I, Multibus II and PC AT

intel

iNA 960 OpenNET NETWORKING SOFTWARE

FULLY COMPLIANT ISO/OSI TRANSPORT AND NETWORK

iNA 960 is a complete Network and Transport (ISO/OSI Layers 3 and 4) software system plus a comprehensive set of network management functions, Data Link (OSI Layer 2) drivers for IEEE 802.3 Ethernet, and system environment features.

FLEXIBLE AND HIGHLY CONFIGURABLE

iNA 960 is a mature, flexible, and ready-to-use software building block for OEM suppliers of networked systems.

This software is highly configurable for designs based on the 82586 and 82588 LAN controllers, 82501 and 82502 Ethernet serial interface and transceiver, and the 80×86 family of microprocessors.

CONFIGURABLE AT THE OBJECT CODE LEVEL

Consisting of linkable object modules, the iNA 960 software can be configured to implement a range of capabilities and interface protocols. iNA 960 has a large installed base and has been used reliably in a variety of systems from IBM PC XT/ATs to VAX/VMS to IBM mainframes.

BASED ON INTERNATIONAL STANDARDS

Based on the ISO/OSI seven layer model for network communications, iNA 960 implements ISO 8073 Transport Class 4 providing reliable full-duplex message delivery service on top of the internet capabilities offered by the network layer. The iNA 960 network layer is an implementation of the ISO 8473 Network Class 3 Connectionless Network Protocol and supports ISO 9542 End System to Intermediate System Network Dynamic Routing. iNA 960 also supports ISO 8602 Connectionless Transport Protocol (Datagram).

PRECONFIGURED iNA 961

iNA 960 contains the preconfigured iNA 961 which includes support for the iSBC 552A and iSBC 186/530.

REMOTE BOOT SERVER SUPPORT

iNA 960 provides basic boot server capabilities that will transmit predefined images to diskless network nodes that request them.

MULTI-SERVER/CONSUMER SUPPORT

iNA 960 supports the powerful Multibus II feature of multiple host and communications boards. Ideal for LAN load balancing and redundant networks for fault-tolerant systems.

FEATURES

- Certified ISO/OSI Transport and Network Layer Software
- ISO 8072/8073 Transport Class 4
- ISO 8602 Connectionless Transport
- ISO 8348/8473 Connectionless Network
- ISO 9542 End System to Intermediate System Dynamic Routing
- Comprehensive Network Management Functions
- Remote Boot Server for diskless workstations
- Data Link Drivers for iSBC 552A, iSBX 586, iSBC 186/530, and iSBC 186/51

PRODUCT CODES

INA960 Includes iNA 961 on iRMX diskette format

MULTIBUS I OpenNET NETWORKING HARDWARE

iSBC 186/51 MULTIBUS I IEEE 802.3/ETHERNET COMMUNICATION COMPUTER

- 82586 LAN coprocessor for Ethernet/IEEE 802.3 communication
- Two serial interfaces, RS232C and RS422A/RS449 compatible
- 6 MHz 80186 microprocessor
- 128 Kbytes of dual-port RAM expandable on-board to 256 Kbytes
- Sockets for up to 192 Kbytes of JEDEC 28-pin standard memory devices
- Product Code: sSBC18651

CPU	80186 (6 MHz)
RAM (Bytes)	128K/256K dual port
EPROM (Bytes)	192K (27256), 96K (27128), 48K (2764), 4K (2732)
Serial I/O	2 ports, 26-pin 3M connector (RS232C/422A/449)
Ethernet I/O	1 port, 10-pin AMP connector
Baud rates (RS232/422/449)	75–9600 baud (async.) 1.76–38.4K baud (sync.)
Timers	3
Interrupts	9 levels, 28 sources
iSBX Connectors	2
Software Support	iRMX I O.S., iNA 960, iRMX-NET
Power Requirements	
+5V	7.45 A
+12V	40 mA
−12V	40 mA

iSBC/iSXM 552A MULTIBUS I IEEE 802.3/ETHERNET NETWORK INTERFACE ADAPTER

- High Performance IEEE 802.3/Ethernet compatible network front-end processor
- Resident network software can be down-loaded over the bus or the LAN
- On-board diagnostic and boot firmware
- iSXM 552A version is a preconfigured controller for executing iNA 961 (ISO 8073 Transport and ISO 8473 Network software)
- Product Code: pSBC552A, pSXM552A

CPU	80186 (8 MHz)
LAN Coprocessor	82586
RAM (Bytes)	256K
EPROM (Bytes)	128K (27512), 16K (2764)
Multibus Address	Any 64 KB boundary with a 16 MB address space
Software Support	iNA 960/961
Power Requirements	
+5V	6.2 A
+12V	0.5 A

intel

MULTIBUS II OpenNET NETWORKING HARDWARE

iSBC 186/530 MULTIBUS II IEEE 802.3/ETHERNET NETWORK INTERFACE ADAPTER

- Provides Ethernet (IEEE 802.3) compatible networking capability for all Multibus II systems
- Multibus II iPSB (Parallel System Bus) interface with full Message Passing capability
- Resident firmware to support Built-in-Self-Test (BIST) power-up diagnostics, and host-to-controller software download
- Four 28-pin JEDEC sites, expandable to 8 sites with iSBC 341 MULTIMODULE for a maximum of 512 Kbytes of EPROM
- Provides one RS232C serial port for use in debug and testing
- Product Code: pSBC186530

CPU	80186 (8 MHz)
DRAM	512 KB on-board
EPROM	4 28-pin JEDEC sites, up to 256 KB (max.) using 27512 devices, up to 512 KB (total) using iSBC 341 module
Ethernet I/O	1 channel 15-pin connector
Controller	82586 LAN Coprocessor
Serial I/O	1 channel RS232C, 15-pin connector
Controller	8031
Leads Supported	TD, RD
Timers	3
Interrupts	5 levels with 5 on-board sources and 255 sources from iPSB Bus
Power Requirements	
+5V	8.8 A (excludes power for user-installed memory devices)
+12V	50 mA
−12V	50 mA

PC BUS OpenNET NETWORKING HARDWARE

PCL2 AND PCL2A NETWORK INTERFACE ADAPTERS (PCL2NIA, PCL2ANIA)

- Intelligent high performance hardware with on-board microprocessor, 16 Kbytes EPROM and 256 Kbytes RAM
- Full slot PC AT, PC XT (or compatible computer system) board
- 80186 microprocessor, 82586 LAN coprocessor, 8 MHz zero-wait-state memory access
- RAM shared by the PC host and PCL2 board via an 8K memory window
- DRAM 512 KB (dual port), 0 wait-state memory access by the CPU
- Jumper selection for Ethernet or IEEE 802.3
- Effective self diagnostics
- Product Code: PCL2NIA

CPU	80186 (8 MHz)
LAN Communications Controller	82586
Ethernet Interface	15-pin connector, 82501 serial interface
DRAM	256 KB (dual-port), 0 wait-state memory access by the CPU
EPROM	16 KB
Size	4.15 in H×13.3 in W
Power Requirements	
+5V	2.0 A
+12V	0.5 A

OpenNET NETWORKING ACCESSORIES

iSBX 586 DATA LINK ENGINE MULTIMODULE BOARD

- Provides an IEEE 802.3/Ethernet compatible connection for 8086 and 80186-based host boards over a 16-bit iSBX interface
- Single-wide iSBX MULTIMODULE
- Compatible with iNA 960/961 ISO 8073 Transport and ISO 8473 Network software
- Provides an IEEE 802.3 to IEEE 802.4 Router capability when used with the iSBC 554 IEEE 802.4 LAN controller
- Product Code: sSBX586

LAN Coprocessor	82586 (8 MHz)
RAM (Bytes)	16K (dual-port)
Software Support	iNA 960/961
Power Requirements	
+5V	2.0 A
+12V	1.0 A

intel

OpenNET NETWORKING ORDERING INFORMATION

Code	Description

iRMX-NET OpenNET PRODUCTS

RMXINETSW	iRMX-NET for iRMX I operating system
RMXIINETSW	iRMX-NET for iRMX II operating system on MB-I, MB-II, and System 120
RMXIIINETSW	iRMX-NET for iRMX III operating system on MB-II and System 120
sSXM120NETKIT	Preconfigured iRMX-NET and sPCLINK2 for networking iRMX and DOS on System 120
RMXIINETVT	Virtual Terminal Support for iRMX II
RMXIIINETVT	Virtual Terminal Support for iRMX III

PCLINK2 OpenNET PRODUCTS

sPCL2NIA*	PCLink2 Network Interface Adapter Hardware only (256 KB)
sPCL2ANIA*	PCLink2 Network Interface Adapter Hardware only (512 KB)
sPCLINK2DEVKIT*	NetBIOS Developer's Kit, NetBIOS programmer kit
MSNET	OpenNET for MS-DOS (order with sPCL2NIA or sPCLANIA)
PCLINKSWOPKIT	OpenNET for MS-DOS (stand-alone software order)

VAX/VMS OpenNET PRODUCTS

VMSNETVA	VMSNET for 11/78x thru 8700 (includes Digital's VOTS)
VMSNETVB	VMSNET for 6xxx thru 89xx (includes Digital's VOTS)
MVMSNETVA	MVMSNET for uVAX 11 and 2000 (includes Digital's VOTS)
MVMSNETVB	MVMSNET for uVAX 3xxx (includes Digital's VOTS)
VMSNETVASW	VMSNET for 11/78x thru 8700 (does not include Digital's VOTS)
VMSNETVBSW	VMSNET for 6xxx thru 89xx (does not include Digital's VOTS)
MVMSNETVASW	VMSNET for uVAX 11 and 2000 (does not include Digital's VOTS)
MVMSNETVBSW	VMSNET for uVAX 3xxx (includes Digital's VOTS)
VMSNETCLIC	VAX Cluster software license for VMSNETVx products

UNIX SYSTEM V OpenNET PRODUCTS

SVNET	SV-OpenNET for UNIX System V Release 3.2 on Multibus I, Multibus II and PC AT

iNA 960 OpenNET ISO/OSI PRODUCTS

INA960JRO	ISO/OSI Transport and Network layers, includes iNA961
INA960RF	Royalty fee for INA960

MULTIBUS I AND MULTIBUS II IEEE 802.3/ETHERNET PRODUCTS

sSBC18651*	iSBC 186/51 Multibus I IEEE 802.3/Ethernet Communication Computer
pSBC552A*	iSBC 552A Multibus I IEEE 802.3/Ethernet Network Interface Adapter
pSXM552A*	iSXM 552A preconfigured for Intel System 310 and 320, includes iNA 961 royalty

OpenNET NETWORKING ORDERING INFORMATION

Code	Description
OpenNET NETWORKING ACCESSORIES	
sSBX586*	iSBX 586 MULTIMODULE IEEE 802.3/Ethernet Data Link board

intel

OpenNET NETWORKING LITERATURE

Code	Description

iRMX-NET OpenNET PRODUCTS

462040	iRMX-NET Software Release 3.1 Installation and Configuration Guide
462041	iRMX-NET Software Release 3.1 User's Guide

PCL2 OpenNET PRODUCTS

450772-002	OpenNET PCLink2 NIA Hardware Reference Manual
459621-002	OpenNET PCLink2 ANIA Hardware Reference Manual
460665-002	OpenNET PCLink Users Guide
460992-001	OpenNET PCLink2 Installation Guide
462311-002	PCLink S/W Developer's Manual (Developer's Kit)
462305-002	PCLink2 Installation Guide
469620-002	PCLink2 ANIA Installation Guide

VAX/VMS OpenNET PRODUCTS

480071	VAX/VMS OpenNET User's Manual

UNIX SYSTEM V OpenNET PRODUCTS

462740	SV-OpenNET User's Manual
462741	SV-OpenNET Installation and Administration Guide

iNA 960 OpenNET ISO/OSI PRODUCTS

467686-001	iNA 960 R3.1 Programmer's Reference Manual
462252-001	iNA 960 R3.1 Installation and Configuration Guide

MULTIBUS I AND MULTIBUS II IEEE 802.3/ETHERNET PRODUCTS

122330	iSBC 186/51 COMMputer Board Hardware Reference Manual
149228	iSBC 552A IEEE 802.3 Communications Controller User's Guide
149226	iSBC 186/530 Network Interface Adapter User's Guide

OpenNET NETWORKING ACCESSORIES

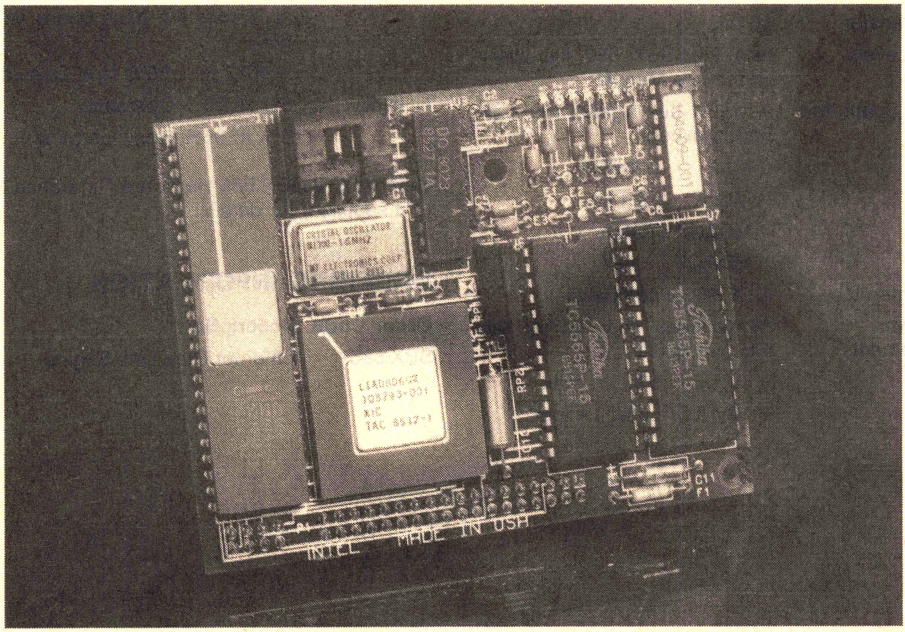
122290	iSBX 586 MULTIMODULE Ethernet Communication Controller Hardware Reference Manual

iSBX 586
Ethernet Data Link Engine

- **Provides an IEEE 802.3 (Ethernet) Data Link Connection for Host Boards with 16-Bit SBX Bus Capability**
- **Based on Intel's 8 MHz 82586 LAN Coprocessor Chip which Includes the following Features:**
  - **— Automatic Retransmission**
  - **— On-Board Multicast Address Filtering**
  - **— Host Interface via Buffer Chaining**
- **16 Kbytes of Local Dual-Ported Buffer RAM**
- **Single-Wide iSBX Board That Conforms to IEEE 959 Specifications**
- **Compatible with iNA 960 ISO Transport Layer Software**
  - **— Direct Support for iRMX® Operating Systems**
  - **— Source Code Support for Other Operating Systems**

The iSBX 586 Ethernet Data Link Engine is a single wide iSBX sized card that provides a low cost Ethernet controller for single board computers with 16-bit SBX bus capability. Based on the 82586 Local Area Network Coprocessor, the iSBX 586 implements the data link (Layer 2) and physical (Layer 1) layers of the International Standards Organization (ISO) Open Systems Interconnect (OSI) Reference Model. This allows the iSBX 586 to supply an IEEE 802.3 10 Mbps (Ethernet) connection for a single board computer having a 16-bit SBX bus connector.

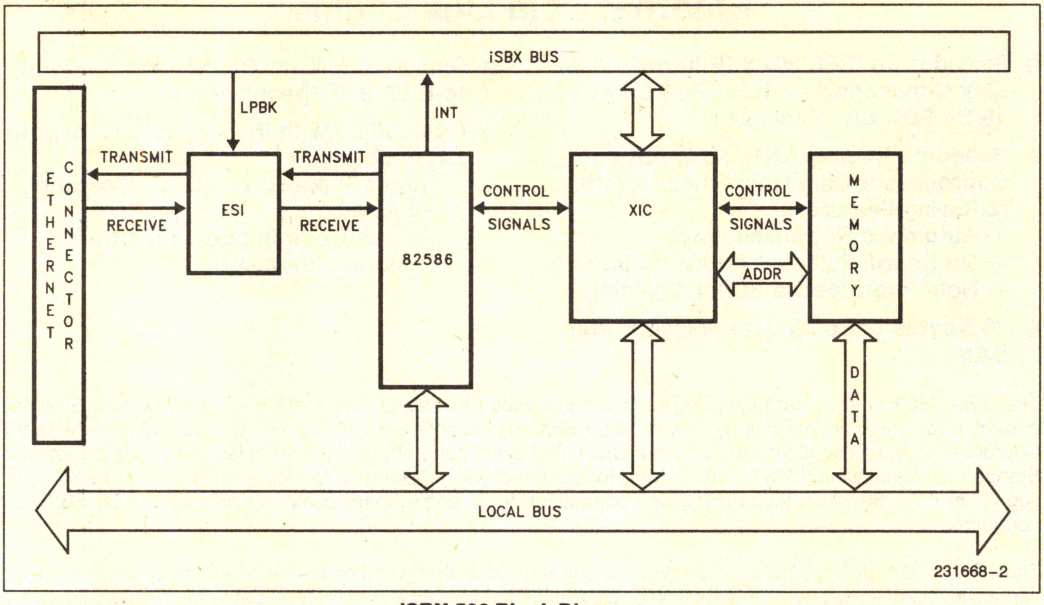
The iSBX 586 MULTIMODULE is a low cost building block that can implement an Ethernet connection at various levels of integration. One application for the iSBX 586 is as a "best effort" datagram message delivery engine. In conjunction with the host single board computer running iNA 960 R2.0 ISO Transport Software, the iSBX 586 can allow for a four-layer, OpenNET compatible solution for Ethernet connections.



231668-1

September 1993
Order Number: 231668-001

BLOCK DIAGRAM



iSBX 586 Block Diagram

SPECIFICATIONS

SBX Interface	IEEE 959
Data Transfer	16 bits
Signal Levels	See the iSBX 586 Hardware Reference Manual
Signals Supported	All iSBX bus signals are supported except: MA2 MINTR1 MCLK OPT1 MDACK TDMA MDRQT −12V
Serial Interface	IEEE 802.3 compatible
DC Power Requirements	All voltages supplied by the iSBX interface +5V DC ±5%, 2A max. +12V DC ±5%, 1A max.

ENVIRONMENTAL

Temperature: 0°C to 55°C Operating (Free moving air across the base board and iSBX 586)
−40°C to +65°C Non-Operating

Humidity: 5% to 90% Operating
5% to 95% Non-Operating

Refer to the iSBX 586 Hardware Reference Manual (not supplied) for details.

ORDERING INFORMATION

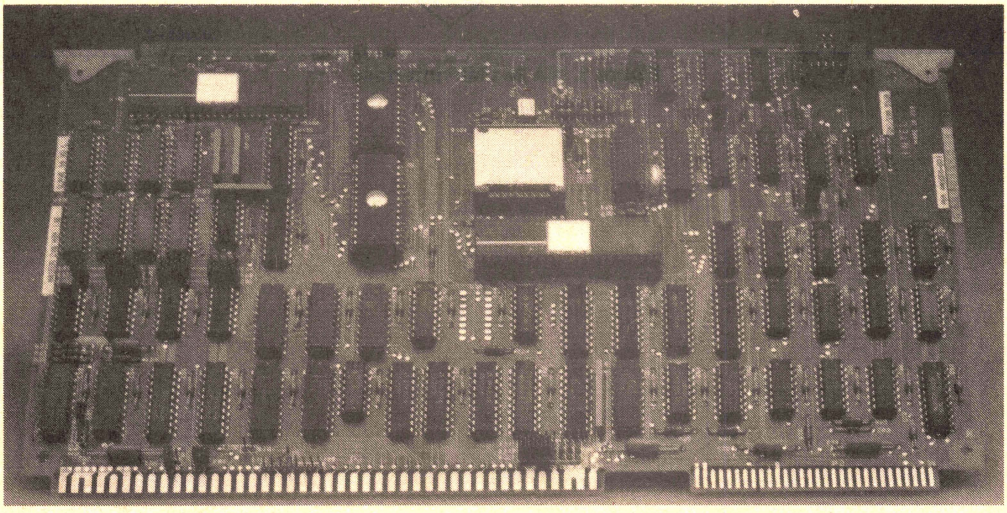
Order Code	Description
SBX 586	Ethernet Data Link Engine

iSBC 552A
IEEE 802.3 COMPATIBLE
NETWORK INTERFACE ADAPTOR

- **Provides High-Performance Network Front-End Processing for All Multibus I Systems Regardless of the Operating System of the Host**
  - **— Intelligent Controller with an 8 MHz 80186 Processor and 256K of DRAM Memory**
  - **— IEEE 802.3 Network Port Driven by the 82586 LAN Coprocessor**
- **Can Execute On-Board the Intel iNA 960/961 Software, an Implementation of Industry Standard ISO 8073 Transport and ISO 8473 Network Protocols**
- **Resident Network Software Can be Down-Loaded over the Bus or LAN**
- **On-Board Diagnositc and Boot Firmware**
- **Supported by XNX-NET and iRMX-NET Network File Service Software Products**
- **iSBC 552A is a Flexible, Intelligent Communications Controller for IEEE 802.3 LANs**

The iSBC 552A is designed for communications front end processor applications connecting Multibus I systems onto IEEE 802.3 compatible LANs. This Network Interface Adaptor is dedicated to the network communications tasks within a system allowing the host to spend more time processing user applications. A major advantage of this product is that it can be used to network existing systems and established designs without forcing the redesign of the entire system architecture.

The iSBC board can be used with any operating system because it requires only a high level interface to communicate with the host. The result is a powerful system building block which enables the OEM to network Multibus I based systems with different operating systems. Applications for the 552A products include networked multiuser XENIX based systems for the office and laboratory, iRMX® software-based systems for real-time applications, or many other system applications.



280385-1

August 1993
Order Number: 280385-002

THE iSBC BOARD

The iSBC 552A version is a board that offers the hardware necessary for the user to construct an IEEE 802.3 front-end processor for custom requirements. The Intel iNA 960 ISO standard transport and network software can be configured and optimized to run on the iSBC 552A board.

Operating Systems Environment

The iSBC 552A board and iNA 960/961 software can function in any Multibus I environment. The communication between the iSBC 552A and the host processor is entirely independent of any host operating systems. iNA 960/961 uses the MIP protocol to interface with the host processor. The MIP is a reliable, host operating system independent, process to process communication scheme between any processors on the Multibus I System Bus. iNA 960/961 can service multiple processes utilizing its services at the same time.

A host processor passes iNA 960/961 commands and buffers in the Multibus I system memory to the iNA 960/961 software. This software is responsible for updating the response fields of these commands. It is responsible for copying the user send buffer in Multibus I system memory into its onboard buffers for transmission and for copying received messages to user buffers in Multibus I system memory.

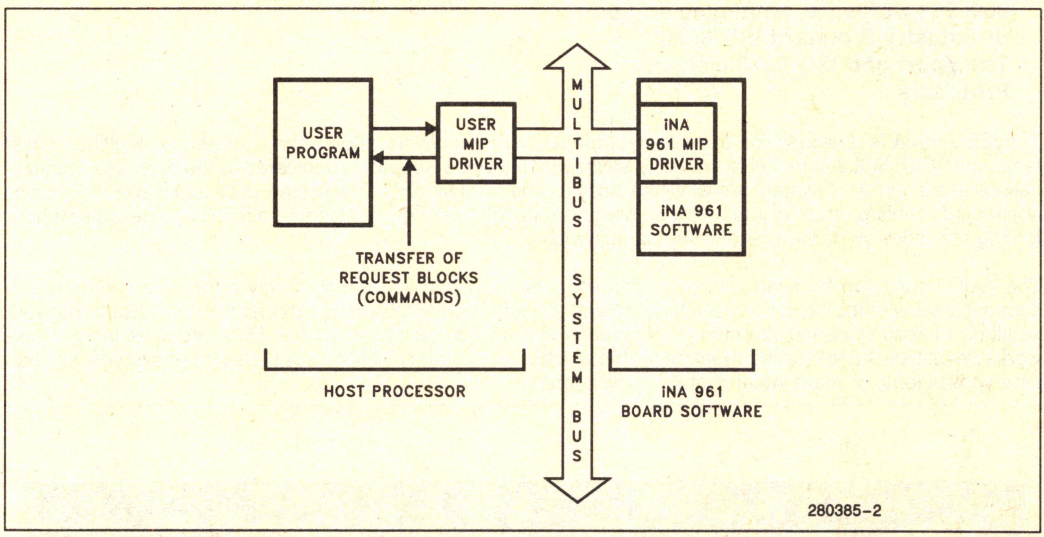


Figure 1. iNA 961 MIP Interface

DEVELOPMENT ENVIRONMENT

The iSBC 552A allows a user to fine tune iNA 960 and to put the software on the board. Both iNA 960 and the iSBC 552A can be flexibly configured to best meet the users' requirements. An Intel development system, together with the Intel I2ICE system or equivalent product can be used if the user desires to do extensive development work on the iSBC 552A. Intel also supplies a wide range of host processor boards and systems (such as the iSBC 286/12 and system 310) that will function well with the iSBC 552A.

SPECIFICATIONS

Data Transfer: 8 or 16 bits

Average Raw Multibus I Transfer Rate:

8.7 Mbits/second (450 ns, 16-bit system memory and no Multibus I contention)

Transceiver Interface

Transmit Data Rate:	10 Mbits/second
Signal Levels:	Series 10,000 ECL-compatible
Host Interrupts:	One Multibus I non-vector interrupt for use in system/host handshaking
Multibus Interface:	The iSBC 552A board conforms to all AC and DC requirements outlined in Intel Multibus I Specification. Order Number 142686-022m except for the following signals: Signal DAT0–DAT7 Signal Specification: I_{IL} = 180 μA I_{IH} = 125 μA
DC Power Required:	All voltages supplied by the Multibus I Interface +5.0V ±5%, 6.2A maximum +12.0V ±5%, 0.5 maximum

Environmental

Temperature:	0°C to +55°C Operating
	−40°C to −65°C Non-Operating
Humidity:	5% to 90% Operating
	5% to 95% Non-Operating

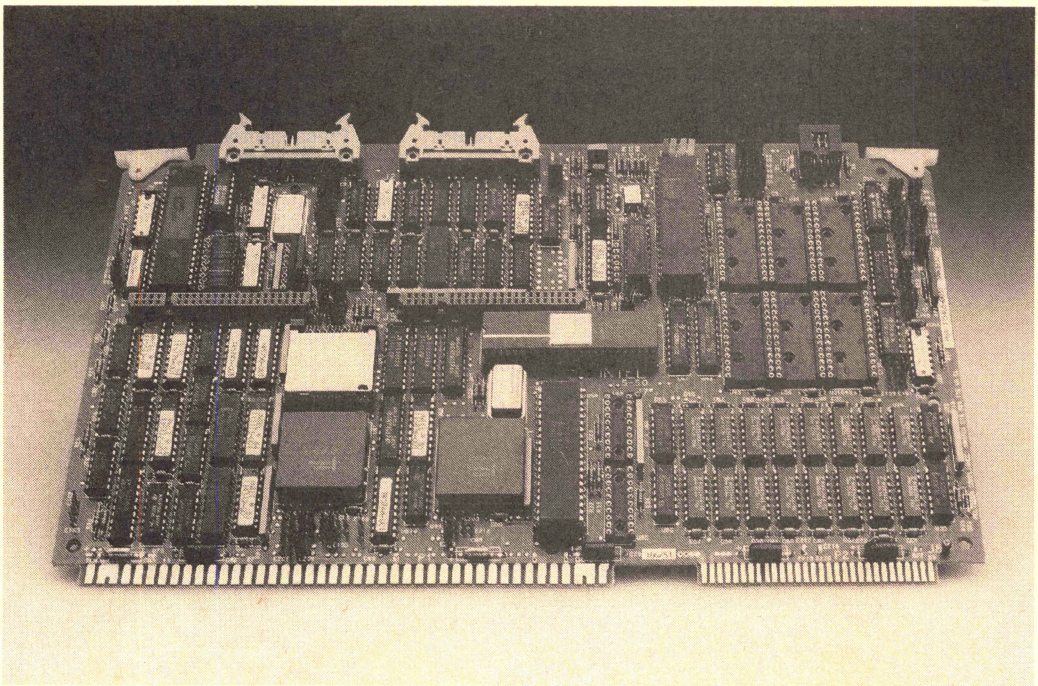
ORDERING INFORMATION

Order Code	Description
SBC552A	IEEE 802.3 COMMengine
NA960	Configurable transport software usable with the SBC552A
NA961	Preconfigured transport software for the SXM552A

iSBC 186/51
COMMUNICATING COMPUTER

- 6 MHz 80186 Microprocessor
- 128K Bytes of Dual-Ported RAM Expandable On-Board to 256K Bytes
- 82586 Local Area Network Coprocessor for Ethernet/IEEE 802.3 Specifications
- Two Serial Interfaces, RS-232C and RS-422A/RS-449 Compatible
- Supports Transport Layer Software (iNA 960) and Higher Layer Communications Software (such as iRMX®-NET)
- Sockets for up to 192K Bytes of JEDEC 28 Pin Standard Memory Devices
- Two iSBX Bus Connectors
- 16M Bytes Address Range of MULTIBUS Memory
- MULTIBUS Interface for Multimaster Configurations and System Expansion
- Supported by a Complete Family of Single Board Computers, Peripheral Controllers, Digital and Analog I/O, Memory, Packaging and Software

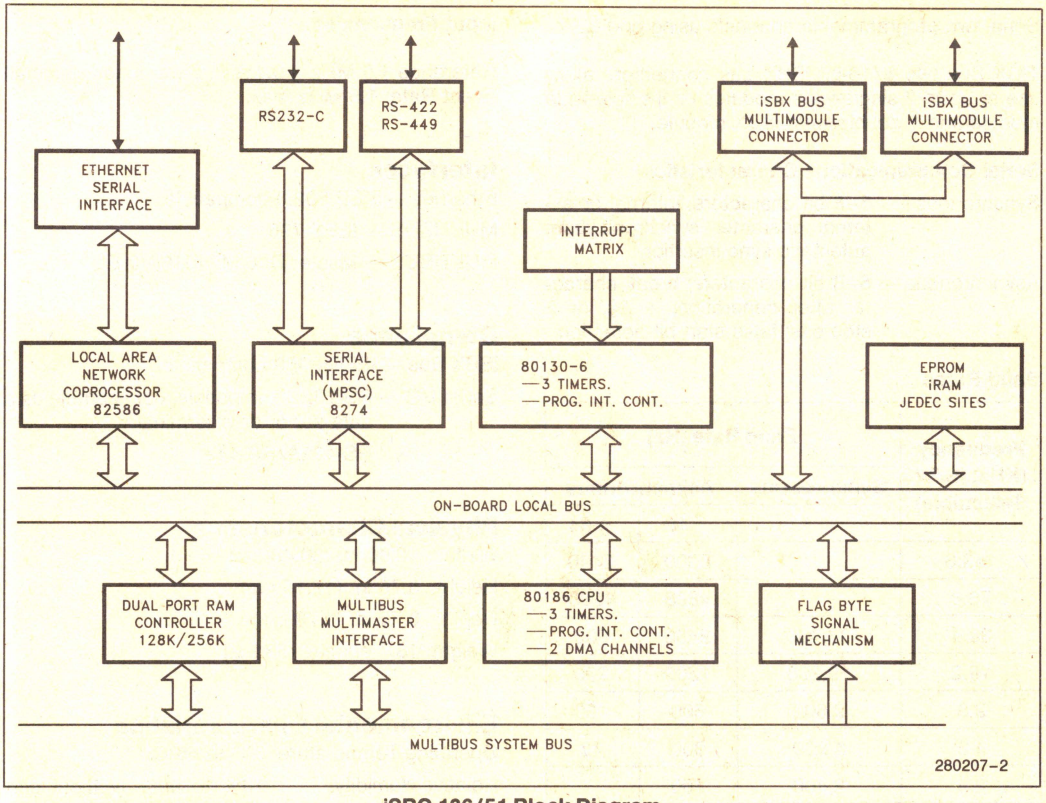
The iSBC 186/51 COMMUNICATING COMPUTER is a member of Intel's OpenNET family of products, and supports Intel's network software. The iSBC 186/51 utilizes Intel's VLSI technology to provide an economical self-contained computer for applications in processing and local area network control. The combination of the 80186 Central Processing Unit and the 82586 Local Area Network Coprocessor makes it ideal for applications which require both communication and processing capabilities such as networked workstations, factory automation, office automation, communications servers, and many others. The CPU, Ethernet interface, serial communications interface, 128K Bytes of RAM, up to 192K Bytes of ROM, I/O ports and drivers and the MULTIBUS interface all reside on a single 6.75″ x 12.00″ printed circuit board.



280207–1

September 1993
Order Number: 280207-001

BLOCK DIAGRAM



iSBC 186/51 Block Diagram

14

SPECIFICATIONS

Word Size

Instruction: 8, 16, 24, or 32 bits
Data: 8, 16 bits

System Clock

6.00 MHz ± 0.1%

Cycle Time

Basic Instruction Cycle

6 MHz— 1000 ns
333 ns (assumes instruction in the queue)

NOTE:
Basic instruction cycle is defined as the fastest instruction time (i.e., two clock cycles).

Memory Capacity/Addressing

Six Universal Memory Sites support JEDEC 24/28 pin EPROM, PROM, iRAM and static RAM.

Example for EPROM:

Device	Total Capacity	Address Range
2732	24K Bytes	F8000–FFFFF$_H$
2764	48K Bytes	F0000–FFFFF$_H$
27128	96K Bytes	E0000–FFFFF$_H$
27256	192K Bytes	C0000–FFFFF$_H$

On-Board RAM

Board	Total Capacity	Address Range
iSBC 186/51	128K Bytes	0–1FFFF$_H$

With MULTIMODULE RAM

Board	Total Capacity	Address Range
iSBC 304	256K Bytes	0–3FFFF$_H$

I/O Capacity

Serial two programmable channels using one 8274.

iSBX Bus two 8/16-bit iSBX bus connectors allow use of up to 2 single-wide modules or 1 single-wide module and 1 double-wide iSBX module.

Serial Communications Characteristics

Synchronous — 5–8 bit characters; internal or external character synchronization; automatic sync insertion

Asynchronous — 5–8 bit characters; break character after generation; 1, ½, or 2 stop bits; false start bit detection

Baud Rates

Frequency (KHz) (S/W Selectable)	Baud Rate (Hz)		
	Synchronous	Asynchronous	
	÷1	÷16	÷64
153.6		9600	2400
76.8		4800	1200
38.4	38,400	2400	600
19.2	19,200	1200	300
9.6	9,600	600	150
4.8	4,800	300	75
2.4	2,400	150	
1.76	1,760	110	2400

NOTE:
Frequency selected by I/O write of appropriate 16-bit frequency factor to baud rate register (80186 timer 0 and 80130 baud timer).

Timers

Input Frequencies

Reference 1.5 MHz ±0.1% (0.5 μs period nominal)
Event Rate: 1.5 MHz max.

Interfaces

Ethernet — IEEE 802.3 compatible

MULTIBUS — IEEE 796

MULTIBUS — Master D16 M24 I16 V0 EL

Compliance

iSBX Bus — IEEE P959 compatible

Serial I/O — RS-232C compatible, configurable as a data set or data terminal,
RS-422A/RS-449

Physical Characteristics

Width: 12.00 in. (30.48 cm)

Height: 6.75 in. (17.15 cm)

Depth: 0.70 in. (1.78 cm)

Weight: 18.7 ounces (531 g.)

Environmental Characteristics

Operating Temperature: 0°C to 55°C

Relative Humidity: 10% to 90% (without condensation)

80186 Output Frequencies/Timing Intervals

Function	Single Timer/Counter		Dual (Cascaded) Timer/Counter	
	Min	Max	Min	Max
Real-Time Interrupt	667 ns	43.69 ms	667 ns	47.72 Minutes
Programmable One-Shot	1000 ns	43.69 ms	1000 ns	47.72 Minutes
Rate Generator	22.889 Hz	1.5 MHz	0.0003492 Hz	1.5 MHz
Square-Wave Rate Generator	22.889 Hz	1.5 MHz	0.0003492 Hz	1.5 MHz
Software Triggered Strobe	1000 ns	43.69 ms	1000 ns	47.72 Minutes
Event Counter		1.5 MHz		

Connectors

Interface	Double-Sided Pins	Centers (In.)	Mating Connectors
Ethernet	10	0.1	AMP87531-5
Serial I/O	26	0.1	3M 3452-0001 Flat or AMP88106-1 Flat

Electrical Characteristics

D.C. Power Supply Requirements

Configuration	Maximum Current (All Voltages ± 5%)		
	+5	+12	−12
SBC 186/51 as shipped:			
Board Total	7.45A	40 mA	40 mA
With separate battery back-up	6.30A	40 mA	40 mA
Battery back-up	1.15A	—	—
With SBC-304 Memory Module Installed:			
Board Total	7.55A	40 mA	40 mA
With separate battery back-up	6.30A	40 mA	40 mA
Battery back-up	1.25A	—	—

NOTES:
1. Add 150 mA to 5V current for each device installed in the 6 available Universal Memory Sites.
2. Add 500 mA to 12V current if Ethernet transceiver is connected.
3. Add additional currents for any SBX modules installed.

14

Reference Manual

122330— iSBC 186/51 Hardware Reference Manual (NOT SUPPLIED)

ORDERING INFORMATION

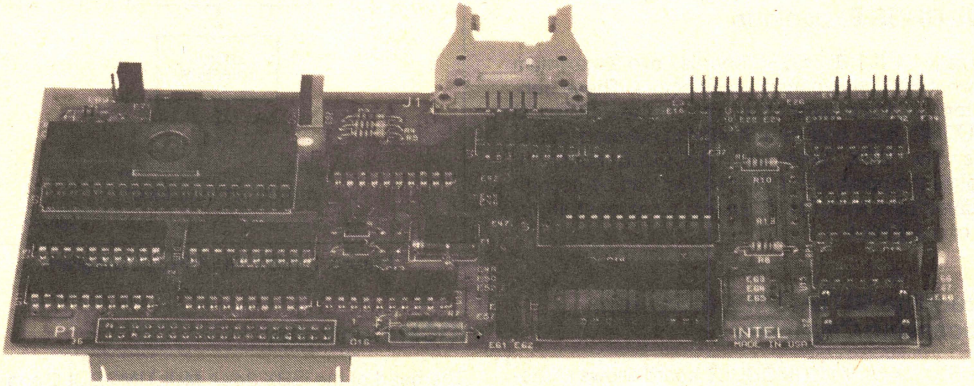
Order Code	Description
SBC18651	Communicating Computer

Distributed Control Modules 15

iSBX 344A
BITBUS INTELLIGENT MULTIMODULE BOARD

- **High Performance 12 MHz 8044 Controller**
- **Integral Firmware Including the iDCX 51 Executive Optimized for Real-Time Control Applications**
- **Full BITBUS Support**
- **2 28-Pin JEDEC Memory Sites for User's Control Functions**
- **Low Cost, Double-Wide iSBX BITBUS Expansion MULTIMODULE Board**
- **Power Up Diagnostics**

The iSBX 344A BITBUS Intelligent MULTIMODULE board is the BITBUS gateway to all Intel products that support the iSBX I/O Expansion Interface. Based on the highly integrated 8044 component (an 8-bit 8051 microcontroller and an SDLC-based controller on one chip) the iSBX 344A MULTIMODULE board extends the capability of other microprocessors via the BITBUS interconnect. With the other members of Intel's Distributed Control Modules (iDCM) family, the iSBX 344A MULTIMODULE board expands Intel's OEM microcomputer system capabilities to include distributed real-time control. The iSBX 344A MULTIMODULE board includes many features that make it well suited for industrial control applications such as: data acquisition and monitoring, process control, robotics, and machine control.



280247-1

15

OPERATING ENVIRONMENT

Intel's Distributed Control Modules (iDCM) product family contains the building blocks to implement real-time distributed control applications. The iDCM family incorporates the BITBUS interconnect to provide standard high speed serial communication between microcontrollers. The iDCM hardware products: including the iSBX 344A MULTIMODULE board, iPCX 344A board and all iRCB BITBUS Remote Controller Boards communicate in an iDCM system via the BITBUS interconnect as shown in Figure 1.

As a member of the iDCM product line the iSBX 344A MULTIMODULE board fully supports the BITBUS microcontroller interconnect. Typically, the iSBX 344A MULTIMODULE board would be part of a node (master or slave) on the BITBUS interconnect in an iDCM system. The iSBX 344A MULTIMODULE board plugs into any iSBC board with an iSBX connector.

The iSBX 344A MULTIMODULE board is the hardware interface between Intel's MULTIBUS and the BITBUS environment. With this interface the user can harness the capabilities of other Intel microprocessors e.g. 80386, 80286, 80186, 8086, 80188, 8088 in a iDCM system or extend an existing MULTIBUS system with the iDCM family.

MULTIBUS Expansion

Typically, MULTIBUS iSBC boards have a maximum of two iSBX I/O expansion connectors. These connectors facilitate addition of one or two iSBX I/O MULTIMODULE boards with varying numbers of I/O lines. The iSBX 344A MULTIMODULE board increases the number of I/O lines that can be accommodated by a MULTIBUS system by at least an order of magnitude.

Extending BITBUS/iDCM System Processing Capability

The iSBX 344A MULTIMODULE board allows utilization of other processors in a iDCM system to accommodate particular application requirements. The MULTIMODULE board is compatible with any iSBX connector so that any board having a compatible connector can potentially enhance system performance. Intel's DCS100 BITBUS Toolbox Software provides easy to use high performance software interfaces for iSBC boards. The iSBC 86/35, 286/12, and 188/48 boards are a few examples. Custom configurations are also possible with user customized software.

ARCHITECTURE

Figure 1 illustrates the major functional blocks of the iSBX 344A board: 8044 BITBUS Enhanced Microcontroller (BEM), memory, BITBUS microcontroller interconnect, Byte FIFO interface, initialization and diagnostic logic.

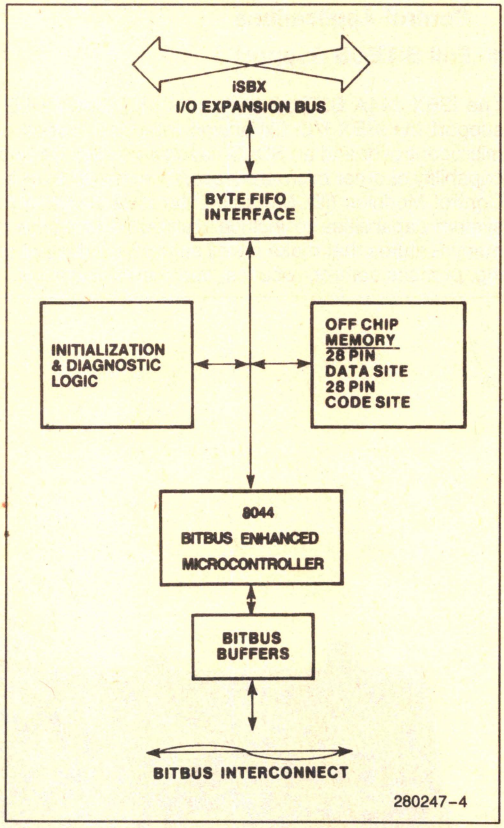


Figure 1. iSBX 344A Block Diagram

iDCM Controller

The heart of the iSBX 344A MULTIMODULE board's controlling and communication capability is the highly integrated 12 MHz 8044 microcontroller. The 8044 consists of the advanced 8-bit, 8051 microcontroller and a SDLC-based controller called the Serial Inter-

face Unit (SIU). This dual processor architecture allows complex control and high speed communication to be realized cost effectively.

The 8044 BEM microcontroller also includes built-in firmware known as DCM44. This firmware includes a set of functions called Remote Access and Control (RAC), a preconfigured version of the DCX51 Executive, communications software, and a power-up test procedure.

Memory

The iSBX 344A MULTIMODULE board memory consists of two internal and external memory. Internal memory is located in the on-chip memory of the iDCM controller. The iDCX 51 Executive and the remaining 8044 BEM firmware ration this resource. However, eight bytes of bit addressable internal memory are reserved for the user. Ample space is reserved for user programs and data in the iSBX 344A MULTIMODULE board external memory.

Two 28-pin JEDEC sites comprise the iSBX 344A MULTIMODULE board external memory. One site has been dedicated for data; the other for code. Table 1 lists the supported memory devices for each site. Intel's 2764 and 27128 are examples. The user may choose one of two memory configurations and specify different memory sizes by placing the proper jumpers at system initialization. The most flexible configuration option provides the user with access to the code site for program download or upload. This feature ensures expansion of an existing system is easily accommodated. For example, the addition of another conveyor to a material handling system would require adding another controller or controllers and changes to existing applications code and addition of new code.

Table 1. Supported Memory Devices

Device	Data Site	Code Site
4K x 8–64K x 8 EPROM/ROM	No	Yes
2K x 8–32K x 8 SRAM	Yes	Yes
2K x 8–16K x 8 NVRAM and E2PROM	No	Yes

BITBUS Microcontroller Interconnect

The iSBX 344A MULTIMODULE board fully supports the BITBUS microcontroller interconnect. The BITBUS interconnect is a serial bus optimized for control applications. The interconnect supports both synchronous and self-clocked modes of operation. These modes of operation are selectable dependent on application requirements as are the transmission rates. Table 2 shows different combinations of modes of operations, transmission rates, and distances. The SDLC-based protocol, BITBUS message format, and compatibility with Intel's other software and hardware products comprise the remainder of this established architecture. These features contribute to BITBUS reliability and usefulness as a microcontroller interconnect.

The BITBUS connection consists of one or two differential pair(s) of wires. The BITBUS interface of the iSBX 344A MULTIMODULE board consists of a half-duplex RS 485 transceiver and an optional clock source for the synchronous mode of operation.

Byte FIFO Interface

The Byte FIFO Interface on the iSBX 344A MULTIMODULE board implements the required hardware buffering between the 8044 BEM and an extension. An extension is defined as a device attached to the iSBX I/O expansion interface on the iSBX 344A MULTIMODULE board. In an iDCM system, an example of an extension is an iSBC 286/12 board which may be considered the host board in a MULTIBUS system. When used with the software handlers in the BITBUS Toolbox, implementation of this interface is complete.

For particular applications, the user may wish to develop a custom software interface to the extension or host board. On the iSBX 344A MULTIMODULE board side of the interface the iDCM firmware auto-

matically accepts messages for the FIFO. No user code is required, increasing the time available for application system development.

The Byte FIFO supports both byte and message transfer protocol in hardware via three register ports: data, command, and status. The extension side supports polled, interrupt, and limited DMA modes of operation (e.g. 80186 type DMA controllers).

Initialization and Diagnostic Logic

Like the other members of Intel's Distributed Control Modules (iDCM) product line, the iSBX 344A MULTIMODULE board includes many features which make it well suited for industrial control applications. Power up diagnostics is just one of these features. Diagnostics simplify system startup considerably, by immediately indicating an 8044 BEM or external bus failure. The LEDs used for power up diagnostics are available for user diagnostics after power up as well as to further contribute to reliable operation of the system.

Initial iSBX 344A MULTIMODULE board parameters are set by positioning jumpers. The jumpers determine the BITBUS mode of operation: synchronous, self-clocked, transmission rate, and address of the iSBX module in the BITBUS system. This minimizes the number of spare boards to be stocked for multiple nodes, decreasing stocking inventory and cost.

INTEGRAL FIRMWARE

Resident firmware located in the 8044 BEM includes: a pre-configured iDCX 51 Executive for user program development; a Remote Access and Control (RAC) function that enables user communication and control of different microcontrollers and I/O points; a communications gateway to connect the BITBUS interconnect, iSBX bus, and iDCX 51 Executive tasks; and power up diagnostics.

The iDCX 51 Executive is an event-driven software manager that can respond to the needs of multiple tasks. This real-time multitasking executive provides: task management, timing, interrupt handling, and message passing services. Table 3 shows the iDCX 51 calls. Both the executive and the communications gateway allow for the addition of up to seven user tasks at each node while making BITBUS operations transparent.

The Remote Access and Control Function is a special purpose task that allows the user to transfer commands and program variables to remote BITBUS controllers, obtain the status of a remote I/O line(s), or reverse the state of a remote I/O line. Table 4 provides a complete listing of the RAC services. No user code need be written to use this function.

The services provided by the iSBX 344A MULTIMODULE board integral firmware simplify the development and implementation of complex real-time control application systems. All iDCM hardware products contain integral firmware thus supplying the user with a total system solution.

Table 2. BITBUS Microcontroller Interconnect Modes of Operation

	Speed Kb/s	Maximum Distance Between Repeaters M/ft	Maximum # Nodes Per Segment	Maximum # Repeaters Between a Master and Any Slave
Synchronous	500–2400	30/100	28	0
Self Clocked	375 62.5	300/1000 1200/4000	28 28	2 10

Segment: Distance between master and repeater or a repeater and a repeater.
Synchronous mode requires user supplied crystal.

Table 3. iDCX 51 Calls

Call Name	Description
TASK MANAGEMENT CALLS	
RQ$CREATE$TASK	Create and schedule a new task.
RQ$DELETE$TASK	Delete specified task from system.
RQGETFUNCTION$IDS	Obtain the function IDs of tasks currently in the system.
INTERTASK COMMUNICATION CALLS	
RQ$ALLOCATE	Obtain a message buffer from the system buffer pool.
RQ$DEALLOCATE	Return a message buffer to the system buffer pool.
RQ$SEND$MESSAGE	Send a message to specified task.
RQ$WAIT	Wait for a message event.
MEMORY MANAGEMENT CALLS	
RQGETMEM	Get available SMP memory.
RQ$RELEASE$MEM	Release SMP memory.
INTERRUPT MANAGEMENT CALLS	
RQ$DISABLE$INTERRUPT	Temporarily disable an interrupt.
RQ$ENABLE$INTERRUPT	Re-enable an interrupt.
RQ$WAIT	Wait for an interrupt event.
TIMER MANAGEMENT CALLS	
RQSETINTERVAL	Establish a time interval.
RQ$WAIT	Wait for an interval event.

Table 4. RAC Services

RAC Service	Action Taken by Task 0
RESET_STATION	Perform a software reset.
CREATE_TASK	Perform an RQ$CREATE$TASK system call.
DELETE_TASK	Perform an RQ$DELETE$TASK system call.
GET_FUNCTION_ID	Perform an RQGETFUNCTION$IDS call.
RAC_PROJECT	Suspend or resume RAC services.
READ_I/O	Return values from specified I/O ports.
WRITE_I/O	Write to the specified I/O ports.
UPDATE_I/O	Update the specified I/O ports.
UPLOAD_MEMORY	Return the values in specified memory area.
DOWNLOAD_MEMORY	Write values to specified memory area.
OR_I/O	OR values into specified I/O ports.
AND_I/O	AND values into specified I/O ports.
XOR_I/O	XOR values into specified I/O ports.
READ_INTERNAL	Read values at specified internal RAM areas.
WRITE_INTERNAL	Write values to specified internal RAM areas.
NODE_INFO	Return device related information.
OFFLINE	Set node offline.
UPLOAD_CODE	Read values from code memory space.
DOWNLOAD_CODE	Write values to specified EEPROM memory.

NOTE:
Internal memory locations are included in the 192 bytes of data RAM provided in the microcontroller. External memory refers to memory outside the microcontroller — the 28-pin sockets of the iSBX 344A module and the iRCB 44/10A board. Each RAC Access Function may refer to multiple I/O or memory locations in a single command.

SPECIFICATIONS

CPU

8044 BITBUS Enhanced Microcontroller (BEM)

Word Size

Instruction: 8 bits

Data: 8 bits

Processor Clock 12 MHz

Instruction Execution Times

1 μs 60% instructions

2 μs 40% instructions

4 μs Multiply & Divide

Memory Capacity/Addressing

iDCM Controller: Up to 64 Kbytes code

Address Range

	Option A	Option B
External Data Memory	0000H–7FFFH	0000H–7FFFH
External Code Memory	1000H–0FFFFH	8000H–0FEFFH
Internal Code Memory	0000H–0FFFH	0000H–0FFFH

Option A: Supports maximum amount of external EPROM code memory.
Option B: Supports downloading code into external RAM or EEPROM memory.

Terminations

Sockets provided on board for ¼ Watt 5% Carbon type resistors. Resistor value to match characteristic impedance of cable as closely as possible—120Ω or greater.

Message Size

54 bytes max

8044 BITBUS Enhanced Microcontroller (8044 + Firmware) I/O Addressing as Viewed from the 8044

Function	Address	Read	Write	Bit	Comments
Data	FF00H	✓	✓		
Command	FF01H	✓	✓		Write sets command to extension — Read clears command from extension
Status					
-RFNF*	B3H	✓		✓	Also INT1 Input
-TFNE*	B2H	✓		✓	Also INT0 Input
-TCMD*	92H	✓		✓	
LED #1	90H	✓	✓	✓	
LED #2	91H	✓	✓	✓	
RDY/NE*	B4H	✓	✓	✓	
Node Address	FFFFH	✓			
Configuration	FFFEH	✓			

iSBX 344A MULTIMODULE Board I/O Addressing as Viewed from the iSBX 344A MULTIMODULE Board

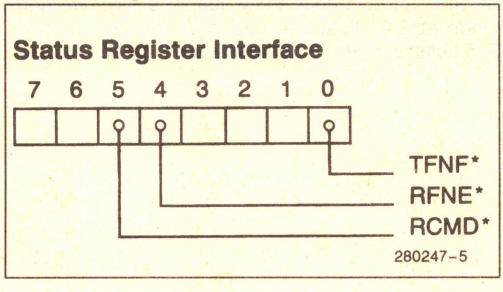
Register Function	Address	Comments
Data	Base′	Read/Write
Command	Base′ + 1	Write sets command from extension Read clears command to extension
Status	Base′ + 2	Read Only

15

Interrupt/DMA Lines

Signal	Location	Interface Option
RINT	MDRQ/MINT0	INT
TINT	MINT1	INT
RCMI	OPT0	INT or DMA
RDRQ	MDRQ/MINT0	DMA
TDRQ	MINT1	DMA

Status Register Interface



Connector Options

10 Pin Plug

Flat Cable: 3M 3473-6010, TB Ansley 609-1001M, or equal

Discrete Wire: BERG 65846-007, ITT Cannon 121-7326-105, or equal

Pinout

Pin	Signal
1	+12V
2	+12V
3	GND
4	GND
5	DATA*
6	DATA
7	DCLK*/RTS*
8	DCLK/RTS
9	RGND
10	RGND

Electrical Characteristics

Interfaces

iSBX I/O Expansion Bus: supports the standard I/O Expansion Bus Specification with compliance level IEEE 959.

Memory Sites: Both code and data sites support the standard 28-pin JEDEC site.

BITBUS Interconnect: Fully supported synchronous mode at 2.4 Mbits/sec and self clocked mode for 375 kbits/sec and 62.5 kbits/sec

The iSBX 344A MULTIMODULE board presents one standard load to the BITBUS bus

Power Requirements

0.9A at +5V ±5% (does not include power to the memory devices)

Physical Characteristics

Double-wide iSBX MULTIMODULE Form Factor

Dimensions

Height: 10.16 mm (0.4 in) maximum component height

Width: 63.5 mm (2.50 in)

Length: 190.5 mm (7.50 in)

Weight: 113 gm (4 ounces)

Environmental Characteristics

Operating Temperature: 0°C to 55°C at 200 Linear Feet/Minute Air Velocity

Humidity: 90% non-condensing

Reference Manual (NOT Supplied)

148099— iSBX 344A Intelligent BITBUS Interface Board User's Guide

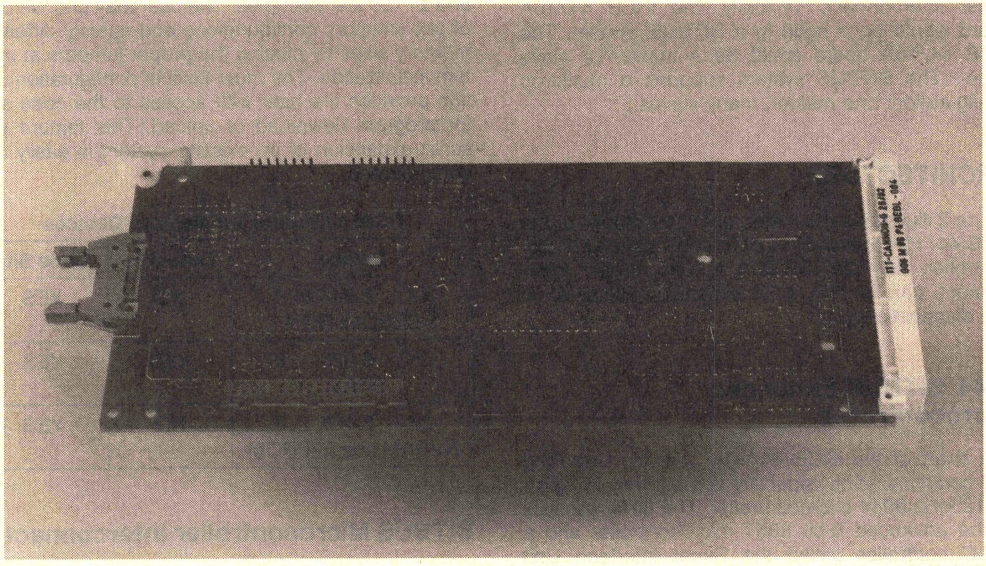
Ordering Information

Part Number	Description
iSBX 344A	BITBUS Intelligent MULTIMODULE board

iRCB 44/10A
BITBUS DIGITAL I/O REMOTE CONTROLLER BOARD

- **High Performance 12 MHz 8044 Controller**
- **Integral Firmware: iDCX Executive, Optimized for Real-Time Control**
- **Full BITBUS Support**
- **Standard Industrial Packaging: Eurocard, DIN Connector**
- **2 28-Pin JEDEC Memory Sites for User's Control Functions**
- **I/O Expansion with 8-Bit iSBX Connector**
- **Programmable Control/Monitoring Using 24 Digital I/O Lines**
- **Power Up Diagnostics**

The iRCB 44/10A BITBUS Digital I/O Remote Controller Board is an intelligent real-time controller and a remote I/O expansion device. Based on the highly integrated 8044 component (an 8 bit 8051 microcontroller and an intelligent SDLC-based controller on one chip) the iRCB 44/10A board provides high performance control capability at low cost. The iRCB 44/10A board can expand Intel's OEM microcomputer system capabilities to include distributed real-time control. Like all members of the iDCM family, the iRCB 44/10A board is well suited for industrial control applications such as data acquisition and monitoring, process control, robotics, and machine control.



280213-1

15

OPERATING ENVIRONMENT

Intel's Distributed Control Modules (iDCM) product family contains the building blocks to implement real-time distributed control applications. The iDCM family incorporates the BITBUS interconnect to provide standard high speed serial communication between microcontrollers. The iDCM hardware products, which include the iPCX 344A board, iSBX 344A MULTIMODULE board and the iRCB 44/10A BITBUS Remote Controller Board (and other iRCB boards), communicate in an iDCM system via the BITBUS interconnect as shown in Figure 1.

The iRCB 44/10A board can be used as an intelligent remote controller or an I/O expansion device. When performing as an intelligent controller the iRCB 44/10A board not only monitors the status of multiple process points, but it can execute varied user supplied control algorithms. When functioning as an I/O expansion device, the iRCB 44/10A board simply collects data from multiple I/O ports and transmits this information via the BITBUS or iSBX bus interface to the system controller for analysis or updating purposes.

As a member of the iDCM product line, the iRCB 44/10A board fully supports the BITBUS microcontroller interconnect. Typically, the iRCB 44/10A board would be a node in a BITBUS system. The iRCB 44/10A board could be a master or slave node. (The BITBUS system supports a multidrop configuration: one master, many slaves.)

ARCHITECTURE

Figure 2 illustrates the major functional blocks of the iRCB 44/10A board: 8044 BITBUS Enhanced Microcontroller, memory, BITBUS microcontroller interconnect, parallel I/O, iSBX expansion, initialization and diagnostic logic.

8044 BITBUS Enhanced Microcontroller

The heart of the iRCB 44/10A board's controlling and communication capability is the highly integrated 12 MHz 8044 microcontroller. The 8044 consists of the advanced 8-bit 8051 microcontroller and a SDLC controller called the Serial Interface Unit (SIU). This dual processor architecture allows complex control and high speed communication functions to be realized cost effectively. The 8044's SIU acts as a SDLC-based controller which offloads the on-chip 8051 microcontroller of communication tasks; freeing the 8051 to concentrate on real-time control.

The 8044 BEM microcontroller also includes, in firmware, a set of procedures known as Remote Access and Control (RAC), a preconfigured version of the DCX 51 Executive, communications software, and power-up diagnostics.

The BEM (8044 microcontroller and on-chip firmware) provides, in one package, a simple user interface, and high performance communications and control capabilities to efficiently and economically build a complex control system.

Memory

The iRCB 44/10A board memory consists of two sections: internal and external. Internal memory is located in the on-chip memory of the BEM. The iDCX51 Executive and the remaining BEM firmware ration this resource. However, eight bytes of bit addressable internal memory are reserved for the user. Ample space is reserved for user programs and data in the iRCB 44/10A board external memory.

Two 28 pin JEDEC sites comprise the iRCB 44/10A board external memory. One site has been dedicated for data, the other for code. Table 1 lists the supported memory devices for each site. Intel's 2764, and 27128 are examples. The user may choose one of two memory configurations and specify different memory sizes by placing the proper jumpers at system initialization. The most flexible configuration option provides the user with access to the code site for program download or upload. This feature ensures expansion of an existing system is easily accommodated.

Table 1. Supported Memory Devices

Device	Data Site	Code Site
4K × 8-64K × 8 EPROM/ROM	NO	YES
2K × 8-32K × 8 SRAM.	YES	YES
2K × 8-16K × 8 NVRAM and E2PROM	NO	YES

BITBUS Microcontroller Interconnect

The iRCB 44/10A board serial interface fully supports the BITBUS microcontroller interconnect. The BITBUS interconnect is a serial bus optimized for control applications. The bus supports both synchronous and self-clocked modes of operation. These modes of operation are selectable dependent on application requirements as are the transmission speeds. Table 2 shows the different combinations of

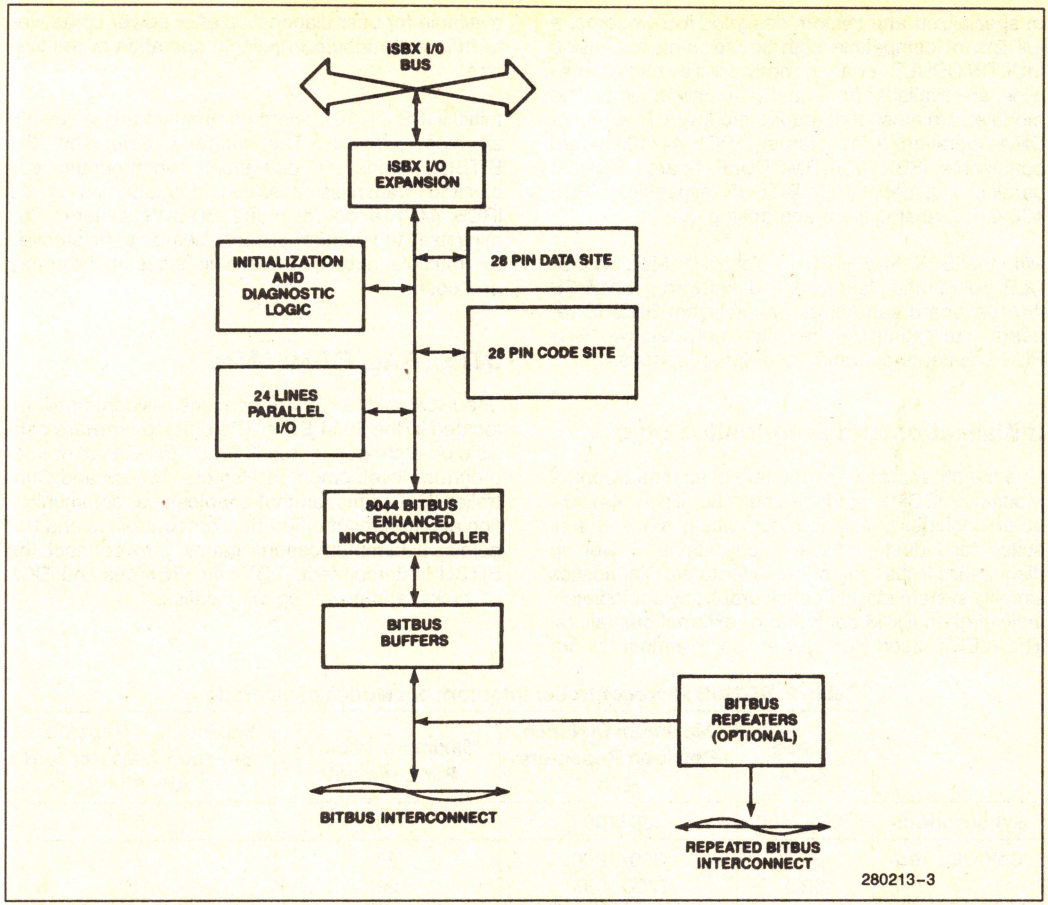


Figure 2. iRCB 44/10A Block Diagram

modes of operation, transmission speeds, and distances. The SDLC-based protocol, BITBUS message format, and compatibility with Intel's other software and hardware products comprise the remainder of the BITBUS architecture. These features contribute to BITBUS system reliability and usefulness as a microcontroller interconnect.

The BITBUS connection consists of one or two differential pair(s) of wires. The serial (BITBUS) interface of the iRCB 44/10A board consists of: a half-duplex RS 485 transceiver, an optional BITBUS repeater and an optional clock source for the synchronous mode of operation.

Digital Parallel I/O

In order to provide an optimal parallel I/O interface for control applications, the iRCB 44/10A board supports 24 software programmable parallel I/O lines. This feature supplies the flexibility and simplicity required for control and data acquisition systems. Sixteen of these lines are fully programmable as inputs or outputs, with loopback, on a bit by bit basis so that bit set, reset, and toggle operations are streamlined. The remaining eight lines are dedicated as inputs. Figure 3 depicts the general I/O port structure.

The parallel I/O lines can be manipulated by using the Remote Access and Control (RAC) function (in BEM firmware) from a supervisory node or locally by a user program. The user program can also access the RAC function or directly operate the I/O lines. Input, output, mixed— input and output, and bit operations are possible simply by reading or writing a particular port.

iSBX Expansion

One iSBX I/O expansion connector is provided on the iRCB 44/10A board. This connector can be used to extend the I/O capability of the board. In addition

to specialized and custom designed iSBX boards, a full line of compatible high speed, 8-bit expansion MULTIMODULE boards, both single and double wide, are available from Intel. The only incompatible modules are those that require the MWAIT* signal or DMA operation. A few of Intel's iRCB 44/10A board compatible iSBX MULTIMODULE boards include: parallel I/O, serial I/O, BITBUS expansion, IEEE 488 GPIB, analog input and analog output.

With the iSBX 344A BITBUS Controller MULTIMODULE board and user supplied software, the iRCB 44/10A board can act as an intelligent BITBUS repeater facilitating the transition between two BITBUS segments operating at different speeds.

Initialization and Diagnostic Logic

Like the other members of Intel's Distributed Control Modules (iDCM) product line, the iRCB 44/10A board includes many features which make it well suited for industrial control applications. Power up diagnostics is just one of these features. Diagnostics simplify system startup considerably, by immediately indicating an iDCM controller or external bus failure. The LEDs used for power up diagnostics are available for user diagnostics after power up as well to further contribute to reliable operation of the system.

Initial iRCB 44/10A board parameters are set by positioning jumpers. The jumpers determine the BITBUS mode of operation: synchronous, self clocked, transmission speed, and address of the iRCB 44/10A board in the BITBUS system. This minimizes the number of spare boards to be stocked for multiple nodes, decreasing stocking inventory and cost.

INTEGRAL FIRMWARE

The iRCB 44/10A board contains resident firmware located in the 8044 BEM. The on-chip firmware consists of: a pre-configured iDCX 51 Executive for user program development; a Remote Access and Controller (RAC) function that enables user communication and control of different microcontrollers and I/O points; a communications gateway to connect the BITBUS interconnect, iSBX bus, iPCX bus and iDCX 51 tasks; and power up diagnostics.

Table 2. BITBUS Microcontroller Interconnect Modes of Operation

	Speed Kb/s	Maximum Distance Between Repeaters M/ft	Maximum # Nodes Per Segment*	Maximum # Repeaters Between A Master And Any Slave
Synchronous	500–2400	30/100	28	0
Self Clocked	375 62.5	300/1000 1200/4000	28 28	2 10

*Segment: Distance between master and repeater or repeater and repeater. Synchronous mode requires user supplied crystal.

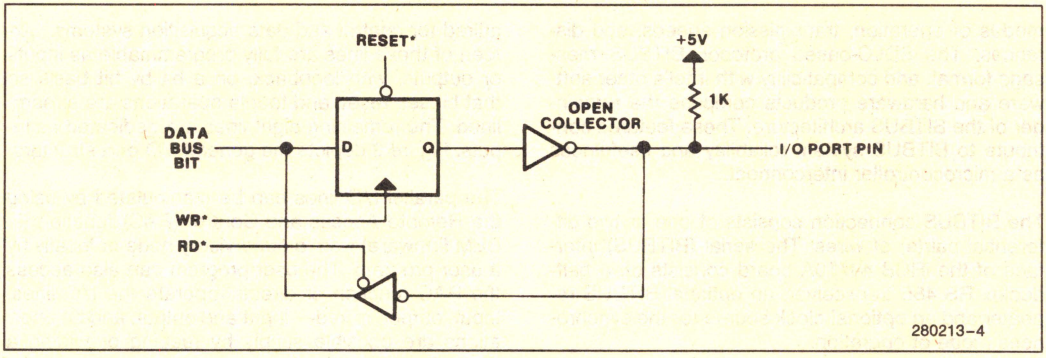


Figure 3. I/O Port Structure

Table 3. iDCX 51 Executive Calls

Call Name	Description
TASK MANAGEMENT CALLS	
RQ$CREATE$TASK	Create and schedule a new task.
RQ$DELETE$TASK	Delete specified task from system.
RQGETFUNCTION$IDS	Obtain the function IDs of tasks currently in the system.
INTERTASK COMMUNICATION CALLS	
RQ$ALLOCATE	Obtain a message buffer from the system buffer pool.
RQ$DEALLOCATE	Return a message buffer to the system buffer pool.
RQ$SEND$MESSAGE	Send a message to specified task.
RQ$WAIT	Wait for a message event.
MEMORY MANAGEMENT CALLS	
RQGETMEM	Get available SMP memory.
RQ$RELEASE$MEM	Release SMP memory.
INTERRUPT MANAGEMENT CALLS	
RQ$DISABLE$INTERRUPT	Temporarily disable an interrupt.
RQ$ENABLE$INTERRUPT	Re-enable an interrupt.
RQ$WAIT	Wait for an interrupt event.
TIMER MANAGEMENT CALLS	
RQSETINTERVAL	Establish a time interval.
RQ$WAIT	Wait for an interval event.

The iDCX 51 Executive is an event-driven software manager that can respond to the needs of multiple tasks. This real-time multitasking executive provides: task management, timing, interrupt handling, and message passing services. Table 3 shows the iDCX 51 calls. Both the Executive and the communications gateway allow for the addition of up to seven user tasks at each node while making BITBUS operation transparent.

The Remote Access and Control Function is a special purpose task that allows the user to transfer commands and program variables to remote BITBUS controllers, obtain the status of a remote I/O line(s), or reverse the state of a remote I/O line. Table 4 provides a complete listing of the RAC services. No user code need be written to use this function. Power up tests provide a quick diagnostic service.

The services provided by the iRCB 44/10A board integral firmware simplify the development and implementation of complex real-time control application systems. All iDCM hardware products contain integral firmware thus supplying the user with a total system solution.

Table 4. RAC Services

RAC Service	Action Taken by Task 0
RESET_STATION	Perform a software reset.
CREATE_TASK	Perform an RQ$CREATE$TASK system call.
DELETE_TASK	Perform an RQ$DELETE$TASK system call.
GET_FUNCTION_ID	Perform an RQGETFUNCTION$IDS call.
RAC_PROTECT	Suspend or resume RAC services.
READ_IO	Return values from specified I/O ports.
WRITE_IO	Write to the specified I/O ports.
UPDATE_IO	Update the specified I/O ports.
UPLOAD_MEMORY	Return the values in specified memory area.
DOWNLOAD_MEMORY	Write values to specified memory area.
OR_I/O	OR values into specified I/O ports.
AND_I/O	AND values into specified I/O ports.
XOR_I/O	XOR values into specified I/O ports.
READ_INTERNAL	Read values at specified internal RAM areas.
WRITE_INTERNAL	Write values at specified internal RAM areas.
NODE_INFO	Return device related information.
OFFLINE	Set node offline.
UPLOAD_CODE	Read values from code memory space.
DOWNLOAD_CODE	Write values to specified EEPROM memory.

INDUSTRIAL PACKAGING

The iRCB 44/10A form factor is a single high, 220 mm deep Eurocard and supports most standard industrial packaging schemes as well as Intel's RCX 910 Digital Signal Conditioning, Isolation and Termination Module (see below). The Eurocard form factor specifies reliable DIN connectors. A standard 64 pin connector is included on the iRCB 44/10A board.

Physical Characteristics

Single high, 220 mm deep Eurocard Form Factor

Dimensions

Width: 13.77 mm (0.542 in) maximum component height

Height: 100 mm (3.93 in.)

Depth: 220 mm (8.65 in.)

Weight: 169 gm (6 ounces)

SPECIFICATIONS

Word Size

Instruction: 8 bits

Data: 8 bits

Processor Clock 12 MHz

Instruction Execution Times

1 μsec 60% instructions
2 μsec 40% instructions
4 μsec Multiply & Divide

Memory Capacity/Addressing

iDCM Controller: Up to 64 Kbytes code

Address Ranges

Memory	Option A	Option B
External —Data	0000H-7FFFH	0000H-7FFFH
—Code	1000H-0FFFFH	8000H-0FEFFH
Internal	0000H-0FFFH	0000H-0FFFH

NOTES:
Option A: Supports maximum amount of external EPROM code memory.
Option B: Supports downloading code into RAM or EEPROM memory.

Interrupt Sources

Two external: iSBX I/O Expansion bus sources or other sources.
BITBUS Microcontroller Interconnect.

8044 BITBUS Enhanced Microcontroller I/O Addressing

Function	Address	Read	Write	Bit
PORT A	FFCOH	✓	✓	
PORT B	FFC1H	✓		
PORT C	FFC2H	✓	✓	
MCSO	FF80H-FF87H FF00, FF01	✓	✓	
MSC1	FF88H-FF8F	✓	✓	
LED #1	90H	✓	✓	✓
LED #2	91H	✓	✓	✓
RDY/NE*	B4H	✓	✓	✓
NODE ADDRESS	FFFFH	✓		
CONFIGURATION	FFFEH	✓		
OPT0	92H	✓	✓	✓
OPT1	93H	✓	✓	✓
INT0	B2H	✓		✓
INT1	B3H	✓		✓

15

PARALLEL I/O

Number: 2 8-Bit Bi-directional Ports
1 8-Bit Input Port

Table 5. Parallel I/O Electrical Specification

Parameter	Condition	Min	Max	Units
V_{OL}	$I_{OL} = 16$ mA		0.5	V
V_{OH}	$I_{OH} = -2$ mA	2.4		V
V_{IH}		2.0	7.0	V
V_{IL}		−1.0	0.8	V
I_{IL}	$V_{IL} = 0.5V$		6.0	mA
I_{IH}	V_{IH} = logic high		0.0	mA
I_I	$V_{IH} = 7V$		−2.2	mA

Terminations

Sockets provided on board for ¼ Watt 5% Carbon type resistors. Resistor value to match characteristic impedance of cable as closely as possible—120Ω or greater.

Repeaters

Sockets provided on board: Devices 75174 and 75175

Connector Options

10 PIN PLUG

Flat Cable: 3M 3473-6010, TB Ansley 609-1001M, or equal

Discrete Wire: BERG 65846-007, ITT Cannon 121-7326-105, or equal

DIN CONNECTOR PLUG

Flat Cable: GW Elco 00-8259-096-84-124, Robinson Nugent RNE-IDC64C-TG30, or equal

Discrete Wire: ITT Cannon G06 M96 P3 BDBL-004 GW Elco 60 8257 3017, or equal

10 Pin Repeater Connector Pin Out

Pin	Signal
1	+12V
2	+12V
3	GND
4	GND
5	DATA*
6	DATA
7	DCLK*/RTS*
8	DCLK/RTS
9	RGND
10	RGND

Electrical Characteristics

Interfaces

iSBX I/O expansion bus: supports the standard I/O Expansion Bus Specification with compliance level D8/8F

Memory Sites: Both code and data sites support the electrical Universal Memory Site specification

BITBUS Interconnect: The iRCB 44/10A Remote Controller Board supports the BITBUS Specification as follows:

Fully supported synchronous mode at 2.4 Mbits/second and self clocked mode for 375 kbits/second and 62.5 kbits/second

The iRCB 44/10A Remote Controller Board presents one standard load to the BITBUS without repeaters, with repeaters two standard loads

Message length up to 54 bytes supported

RAC Function support as shown in Table 4

Parallel I/O: See the Table 5 for Electrical Specifications of the interface.

Power Requirements

0.9A at +5V ±5% iRCB 44/10 board only (power to memory, repeater, or iSBX board NOT included)

Environmental Characteristics

Operating Temperature:	0°C to 55°C at 200 Linear Feet/Minute Air Velocity
Humidity:	90% non-condensing

Reference Manual (NOT Supplied)

iRCB 44/10A Digital I/O Remote 148100-001 Controller Board User's Guide

Ordering Information

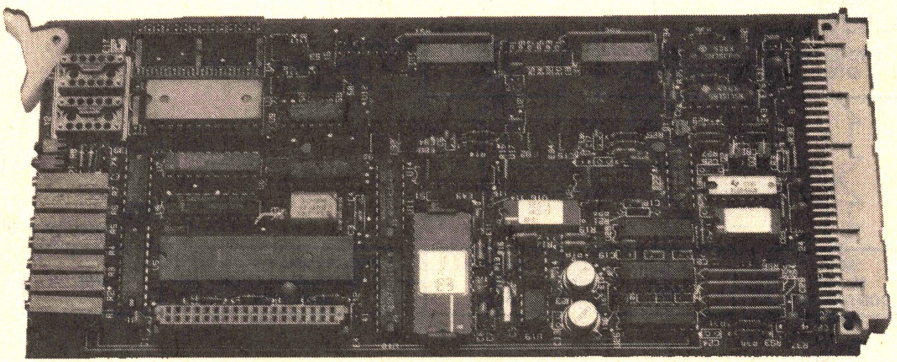
Part Number	Description
RCB4410A	BITBUS Digital I/O Remote Controller Board

iRCB 44/20A
ANALOG I/O CONTROLLER

- **Distributed Intelligence via BITUS Serial Bus**
- **8044 8-bit Microcontroller at 12 MHz**
- **12-bit Analog Resolution**
- **Up To 20 KHz Aquisition Rate (50 ms)**
- **Software Programmable Gain: 1, 10, 100, 500**
- **Two 28-pin JEDEC Memory Sites**
- **16 Single-ended or 8 Differential Input Channels**
- **2 Outputs Channels**
- **±10V Range or 4–20 mA Current Loop**
- **I/O Expandable via iSBX Connector**
- **Compact Single-Eurocard Packaging**
- **Low Power Consumption**
- **Compatible with iRCX 920 Analog Signal Conditioning, Isolation and Termination Panel**

The iRCB 44/20A is a fully programmable analog I/O subsystem on a single-Eurocard form-factor board. The resident 8044 microcontroller operating at 12 MHz provides a means of executing data aquisition and control routines remote from the host computer. Real-time capability is made possible by the iDCX 51 Distributed Control Executive, resident in the 8044 microcontroller. Distribution of real-time control is implemented by the BITBUS Serial Bus protocol, which is also managed integrally by the 8044.

Offering high performance, low-cost, and improved system bandwidth via distributed intelligence, the iRCB 44/20A Analog I/O Controller is ideal for data acquisition and control in both laboratory and industrial environments.



280721-1

IBM PC is a registered trademark of International Business Machines Corporation.

August 1993
Order Number: 280721-003

APPLICATION ENVIRONMENT

Intel's Distributed Control Modules (iDCM) product family contains the building blocks to implement real-time distributed control applications. The iDCM family incorporates the BITBUS interconnect to provide standard high-speed serial communication between microcontrollers. The iRCB 44/20A may communicate with other nodes in a distributed system via the BITBUS interconnect. Other nodes in the system may be the iSBX 344A BITBUS Controller MULTIMODULE, the iPCX 344A BITBUS IBM PC Interface, the iRCB 44/10A BITBUS Digital I/O Controller Board, or other BITBUS compatible products.

The iRCB 44/20A board, can be used as an intelligent remote controller or an I/O expansion device. When performing as an intelligent controller the iRCB 44/20A board not only monitors the status of multiple sensors, it can also locally execute user developed control algorithms. When functioning as an I/O expansion device the iRCB 44/20A board manages the multiple I/O ports, transmitting this information via the BITBUS bus or iSBX interface to the system controller for analysis or data logging purposes.

Typically, the iRCB 44/20A board will operate as a node in a BITBUS system. BITBUS communication supports a multidrop configuration with one master, and multiple subordinate nodes. The iRCB 44/20A board may be either a master or slave node to manage a wide variety of analog input or output tasks.

FUNCTIONAL DESCRIPTION

The major functional blocks of the iRCB 44/20A board, shown in Figure 1, include the 8044 microcontroller and BITBUS interconnect, local memory, Analog I/O, and iSBX expansion.

Distributed Intelligence

The heart of the iRCB 44/20A board's controlling and communication capability is the highly integrated 8044 microcontroller which operates at 12 MHz. The 8044 contains the advanced 8-bit, 8051 microcontroller and a complimentary SDLC controller, called the Serial Interface Unit (SIU). This dual processor architecture provides complex control and high speed communication functions at a low cost.

Another essential part of the 8044 controller is the on-chip firmware that exercises the BITBUS interface. The 8044's SIU acts as an SDLC controller, off loading the on-chip microcontroller of communication tasks so it may concentrate on real-time control.

The 8044 microcontroller simplifies the user interface, and offers high performance communications and control capabilities in a single component package. Many interconnected Distributed Control Modules can form a powerful platform to efficiently and economically administer a complete control system.

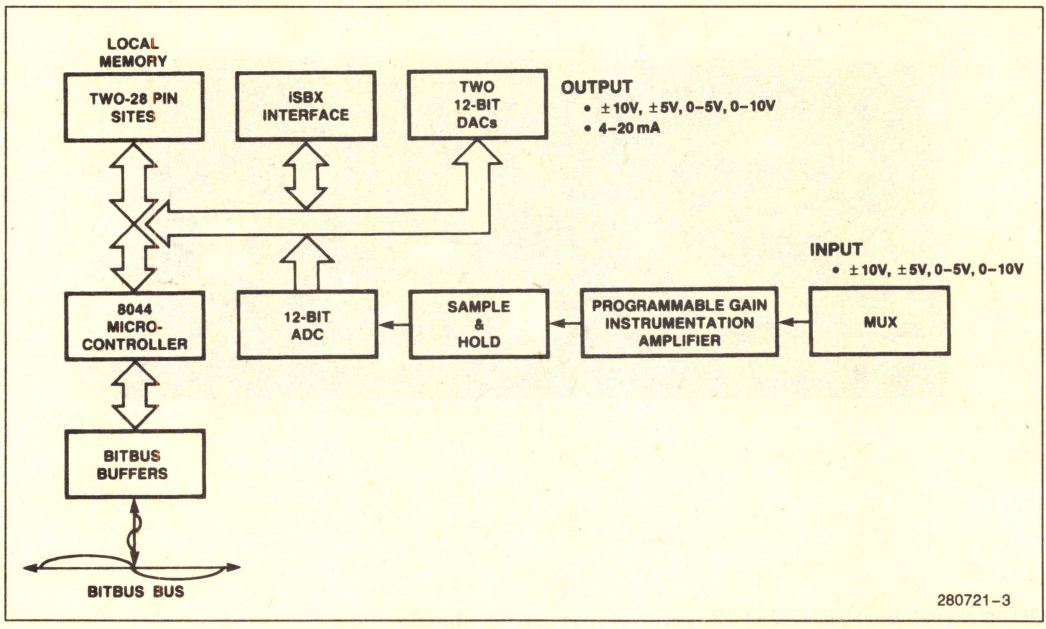


Figure 1. iRCB 44/20A Functional Diagram

Table 1. BITBUS Microcontroller Interconnect Modes Of Operation

	Speed Kb/S	Maximum Distance Between Repeaters M/ft	Maximum # Nodes Per Segment*	Maximum # Repeaters Between A Master And Any Slave
Synchronous	500–2400	30/100	28	0
Self Clocked	375 62.5	300/1000 1200/4000	28 28	2 10

* Segment: Distance between master and repeater or repeater and repeater. Synchronous Mode requires user supplied crystal.

BITBUS Microcontroller Interconnect

The iRCB 44/20A board fully supports the BITBUS microcontroller interconnect. BITBUS is a serial bus optimized for control applications. Both synchronous and self-clocked modes of operation are supported as well as multiple transmission rates. Table 1 shows the different combinations of modes of operation, transmission speeds, and distances. The SDLC protocol and BITBUS message format comprise the data-link level of the BITBUS architecture. Use of these standards maximizes system reliability and flexibility.

The physical connection to BITBUS uses either one or two pairs of wires across which differential signals travel. The iRCB 44/20A board contains a half-duplex RS 485 tranceiver and an optional clock source for the synchronous mode of operation.

Local Memory

The iRCB 44/20A board contains both internal and external local memory. Internal memory is located within the 8044 controller and is used by the iDCX 51 Executive and the SIU. Eight bytes of bit-addressable internal memory have been reserved for the user.

Two 28-pin JEDEC sites provide the iRCB 44/20A board with memory that is external to the 8044. One site has been dedicated for data, the other for application code. Table 2 lists the supported memory devices for each site. The user may select one of two memory configurations using jumpers. One option provides the user with access to the application code site for uploading or downloading programs, which allows expansion or modification of an existing system from a remote site.

Table 2. Supported Memory Devices

Device	Data Site	Code Site
4K x 8–64K x 8 EPROM/ROM	NO	YES
2K x 8–32K x 8 SRAM	YES	YES
2K x 8–16K x 8 NVRAM and E2PROM	NO	YES

Analog I/O

The iRCB 44/20A has been designed to manage a wide variety of analog functions. The jumper-selectable voltage or current ranges plus software programmable gain allows the iRCB 44/20A to acquire data from a combination of up to 16 thermocouples, strain gauges, pressure transducers, flow meters, level sensors, or any devices that operate on a 4–20 mA current loop. Two analog output channels provide the capability to adjust system parameters locally through servo control, voltage-driven devices, or other actuators that respond to 4–20 mA signals.

The 8044 microcontroller on the iRCB 44/20A allows Proportional Integral/Derivative (PID) algorithms, event timing, or averaging tasks to operate independent of the host computer or programmable controller. By off-loading the host in this manner, the overall system performance can be improved significantly.

The analog I/O lines can be manipulated from a remote supervisor by communicating with the Remote Access and Control (RAC) functions, which are included in the 8044 controller firmware. The local application program running on the iRCB 44/20A can also access the RAC functions or directly operate the I/O lines.

iSBX Expansion

One 8-bit iSBX I/O expansion connector is provided to expand the functionality of the iRCB 44/20A board. A full line of compatible expansion MULTIMODULE boards are available from Intel; both single- and double-wide versions are supported by the iRCB 44/20A. Parallel I/O, serial I/O, IEEE 488, magnetic-bubble memory, or additional analog I/O may be added in this manner.

Also, the iSBX 344A BITBUS Controller MULTIMODULE can be used to implement another BITBUS hierarchy with the iRCB 44/20A functioning as the master. With user supplied software, this product

combination can operate as an intelligent BITBUS repeater, facilitating the transmission between two BITBUS segments operating at different speeds.

Initialization and Diagnostic Logic

Like the other members of the Intel's Distributed Control Modules (iDCM) product line, the iRCB 44/20A board includes many features which make it well suited for industrial control applications. Power-up diagnostics simplify system initialization by immediately indicating a failure in either the 8044 microcontroller or external bus. On-board LEDs indicate diagnostic status and are available after power-up for user developed diagnostic routines.

Initial iRCB 44/20A board parameters are manually set with jumpers. These jumpers specify the mode of operation (synchronous or self clocked), and transmission speed. The address of the iRCB 44/20A board within the BITBUS system is also declared in this manner. Therefore, spare board inventory is reduced, since the iRCB 44/20A may be positioned at any node address.

INTEGRAL iDCX 51 FIRMWARE

The iRCB 44/20A board contains resident firmware located within the 8044 controller. This on-chip firmware, known as DCM 44, consists of a pre-configured iDCX 51 Distributed Control Executive for user program development and execution, a library of Remote Access and Control (RAC) functions for inter-node communications and I/O control, plus an iSBX communications gateway, and power-up diagnostics.

The iDCX 51 Executive is an event-driven software manager that can respond to the needs of multiple tasks. This real-time multitasking executive provides task management and timing, interrupt handling, and message passing services. Table 3 shows the iDCX 51 user command library. Both the executive and the communications gateway allow for the addition of seven user tasks at each node that are independent of BITBUS bus management operations.

Remote Access and Control (RAC) functions are special purpose tasks that allow the host system to transfer commands and program variables to remote BITBUS controllers and read/write to the remote I/O lines. Table 4 provides a complete listing of the RAC commands. No user code need be written to use this function. Power-up tests provide a quick diagnostic service.

The DCM 44 firmware, integral to the iRCB 44/20A board, simplifies the development and implementation of complex real-time control applications. All iDCM hardware products contain this integral firmware, providing the user with application code portability.

Table 3. iDCX 51 Executive Calls

Call Name	Description
TASK MANAGEMENT CALLS	
RQ$CREATE$TASK	Create and schedule a new task.
RQ$DELETE$TASK	Delete specified task from system.
RQGETFUNCTION$IDS	Obtain the function IDs of tasks currently in the system.
INTERTASK COMMUNICATION CALLS	
RQ$ALLOCATE	Obtain a message buffer from the system buffer pool.
RQ$DEALLOCATE	Return a message buffer to the system buffer pool.
RQ$SEND$MESSAGE	Send a message to specified task.
RQ$WAIT	Wait for a message event.
MEMORY MANAGEMENT CALLS	
RQGETMEM	Get available SMP memory.
RQ$RELEASE$MEM	Release SMP memory.
INTERRUPT MANAGEMENT CALLS	
RQ$DISABLE$INTERRUPT	Temporarily disable an interrupt.
RQ$ENABLE$INTERRUPT	Re-enable an interrupt.
RQ$WAIT	Wait for an interrupt event.
TIMER MANAGEMENT CALLS	
RQSETINTERVAL	Establish a time interval.
RQ$WAIT	Wait for an interval event.

Table 4. RAC Services

RAC Service	Action Taken by Task 0
RESET_STATION	Perform a software reset.
CREATE_TASK	Perform an RQ$CREATE$TASK system call.
DELETE_TASK	Perform an RQ$DELETE$TASK system call.
GET_FUNCTION_ID	Perform an RQGETFUNCTION$IDS call.
RAC_PROTECT	Suspend or resume RAC services.
READ_IO	Return values from specified I/O ports.
WRITE_IO	Write to the specified I/O ports.
UPDATE_IO	Update the specified I/O ports.
UPLOAD_MEMORY	Return the values in specified memory area.
DOWNLOAD_MEMORY	Write values to specified memory area.
OR_I/O	OR values into specified I/O ports.
AND_I/O	AND values into specified I/O ports.
XOR_I/O	XOR values into specified I/O ports.
READ_INTERNAL	Read values at specified internal RAM areas.
WRITE_INTERNAL	Write values to specified internal RAM areas.
NODE_INFO	Return device related information.
OFFLINE	Set node offline.
UPLOAD_CODE	Read values from code memory space.
DOWNLOAD_CODE	Write values to specified EEPROM memory.

NOTES:
Internal memory locations are included in the 192 bytes of data RAM provided in the microcontroller. External memory refers memory outside the microcontroller—the 28-pin sockets of iRCB 44/20A board. Each RAC Access Function may refer to 1, 2, 3, 4, 5 or 6 individual I/O or memory locations in a single command.

INDUSTRIAL PACKAGING

The iRCB 44/20A board conforms to a single-wide (3V), 220 mm deep Eurocard form-factor. This allows the iRCB 44/20A to fit within standard industrial racks or chassis as well as Intel's RCX 920 Analog Signal Conditioning, Isolation & Termination Panel (see below). The Eurocard specification references DIN 41612 connectors, which are used on the iRCB 44/20A board.

SPECIFICATIONS

CPU

8044 BITBUS Enhanced Microcontroller (BEM)

Word Size

Instruction—8 bits
Data—8 bits

Processor Clock

12 MHz

Instruction Execution Times

1 μsec 60% instructions
2 μsec 40% instructions
4 μsec Multiply & Divide

Memory Addressing

iDCM Controller
Up to 64K bytes code

Address Ranges

	Option A	Option B
External Data Memory Site	0000H–7FFFH	0000H–7FFFH
External Code Memory Site	1000H–0FFFFH (0000H–0FFFFH if EA Active)	8000H–0FFEFH
Internal Code Memory	0000H–0FFFH	0000H–0FFFH

NOTES:
Option A: Supports maximum amount of external EPROM code memory
Option B: Supports downloading code into external RAM or EEPROM memory

I/O Capability

Analog—16 single-ended or 8 differential channels and 2 outputs channels

Expansion—one single-or double-wide iSBX MULTIMODULE (MWAIT * or DMA not supported by iRCB 44/20)

Interrupt Sources

Two external: iSBX I/O Bus or BITBUS Interconnect sources

Bus Termination

Jumper selectable resistors provide termination capability for cable with an impedance of 120Ω or greater.

Analog Input Specifications

Number of channels—16 single-ended or 8 differential
Input ranges—0 to 5V, 0 to 10V (unipolar) ±5V, ±10V (bipolar)
Gain ranges—1, 10, 100, 500, (software programmable)
Input impedance—100MΩ
Input bias current—±50 nA
Overvoltage protection—±32V power on
±20V power off

Accuracy

Resolution—12 bits
Linearity and Noise—±¾ LSB (trimmable)
System Accuracy
Gain = 1—±0.035% full-scale range (trimmable)
Gain = 500—±0.15% full-scale range (trimmable)

Stability

Gain tempco—32 ppm/°C (gain = 11)
75 ppm/°C (gain = 500)
Offset tempco—100 microvolts/°C max.

Dynamic Performance

Aggregate throughout—20 KHz (gain = 1, 10)
7.5 KHz (gain = 100, 500)
Common mode rejection—70 dB (gain = 1)
100 dB (gain = 500)
A/D conversion time—30 microseconds

Analog Output Specifications

Number of channels—2
Output ranges—0 to 5V, 0 to 10V (unipolar) ±5V, ±10V (bipolar)
Current-loop range—4 to 20 mA (unipolar mode only)
Output impedance—0.2Ω min. (voltage)
5 MΩ max. (current)
Output current—±5 mA (short-circuit protected)

Accuracy

Resolution—12 bits
Linearity and Noise—±¾ SB (trimmable)
System Accuracy—
Gain = 1—−0.35% full-scale range (trimmable)
Gain = 500—±0.15% full-scale range (trimmable)

Stability

Full-scale temperature coefficient
150 microvolts/°C (unipolar)
300 microvolts/°C (bipolar)
0.6 microamps/°C (current-loop)
Offset temperature coefficient
30 microvolts/°C (unipolar)
180 microvolts/°C (bipolar)
0.3 microamps/°C (current-loop)

Mating Connectors

Function	# of Pins	Type	Vendor	Part Number
BITBUS Connector	64	Flat Cable	GW Elco Robinson Nugent	00-8259-096-84-124 RNE-IDC-64C-TG30
iSBX Connector	36	Wire Wrap	ITT Cannon GW Elco	G06 M96 P3 BDBL-004 60 8257 3017
		Solder	Viking	000292-0001

Dynamic Performance

Aggregate throughput—20 KHz (gain = 1, 10)
7.5 KHz (gain = 100, 500)
Settling Time—15 microseconds to ±½ LSB

Electrical Characteristics

Interface Compliance

iSBX BUS (through level D8/8F):
Memory sites—code and data sites are JEDEC compatible

BITBUS:

— Synchronous and self-clocked mode support for 500 Kbps to 2.4 Mbps, 375K and 62.5K bits/sec

NOTE:
On-board ALE clock supports 1 Mbps synchronous operation. All other synchronous mode speeds require user-supplied 2.0–9.6 MHz crystal.

— Equivalent to 1.1 standard (RS 485) loads

— Message length up to 54 bytes maximum

Power Requirement (exclusive of optional memory or iSBX MULTIMODULE)

Voltage	Current (amps)	Max, Power (watts)
+5V ±5% +12V ±5% −12V ±5%	0.9 max. 0.7 typ 100 mA max. 100 mA max.	4.5

NOTE:
+15V and −15V required for 0 to 10V and ±10V ranges; for ±15V operation, the iRCB 44/20A cannot be used with iSBX MULTIMODULES that use ±12V power sources.

Physical Characteristics

Width: 3.77 mm (0.542 in) maximum component height
Height: 100 mm (3.93 in)
Depth: 220 mm (8.65 in)
Weight: 169 gm (6 ounces)

Environmental Characteristics

Operating Temperature: 0°C to +60°C at 0.8 CFM air volume
Relative Humidity: 90% non-condensing

Reference Manual (Not Supplied)

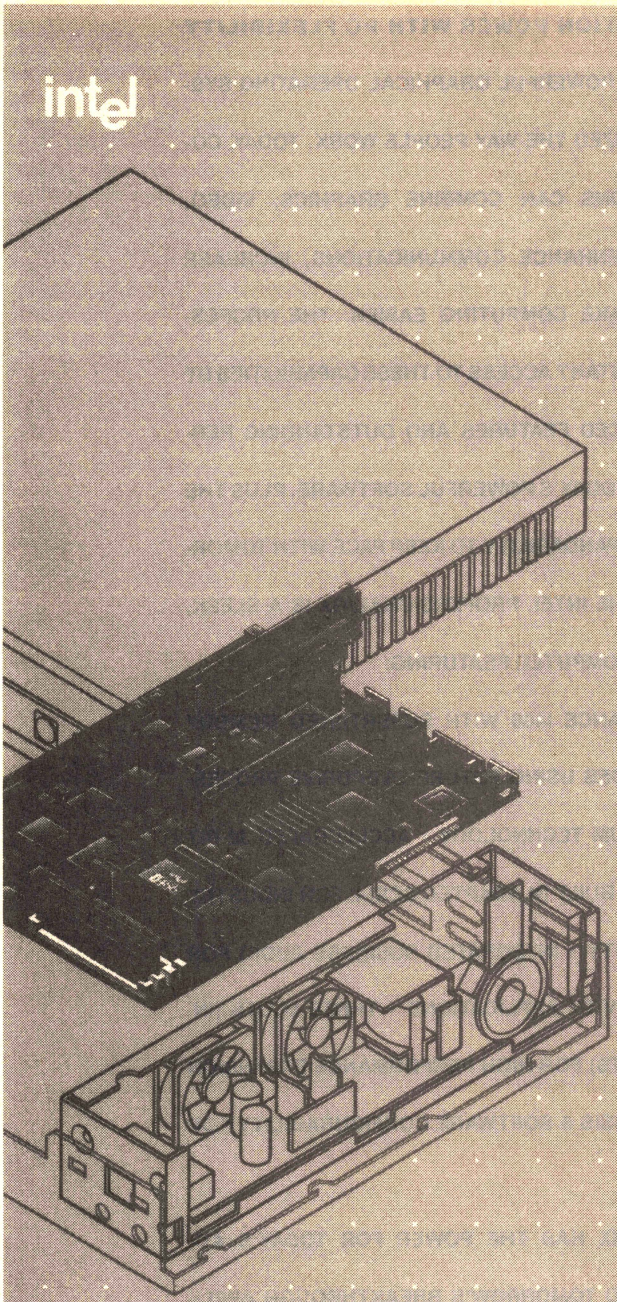
148816— iRCB 44/20A Hardware Reference Manual

ORDERING INFORMATION

Part Number	Description
iRCB 44/20A	BITBUS Analog I/O Controller Board

15

Microcomputer Platforms 16



INTEL486™ PROFESSIONAL/GX WORKSTATIONS

16

October 1993
Order Number: 281555-001

GRAPHICS WORKSTATION POWER WITH PC FLEXIBILITY

A NEW GENERATION OF POWERFUL GRAPHICAL OPERATING SYSTEMS HAS REVOLUTIONIZED THE WAY PEOPLE WORK. TODAY, CO-OPERATING APPLICATIONS CAN COMBINE GRAPHICS, VIDEO, AUDIO AND TEXT TO ENHANCE COMMUNICATIONS, INCREASE PRODUCTIVITY, AND MAKE COMPUTING EASIER. THE PROFESSIONAL/GX DELIVERS INSTANT ACCESS TO THESE CAPABILITIES BY PROVIDING THE ADVANCED FEATURES AND OUTSTANDING PERFORMANCE NEEDED BY TODAY'S POWERFUL SOFTWARE, PLUS THE UPGRADABILITY AND EXPANDABILITY TO KEEP PACE WITH TOMORROW'S TECHNOLOGY. THE INTEL PROFESSIONAL/GX IS A SLEEK, COMPACT, INNOVATIVE COMPUTER FEATURING:

- HIGHEST PERFORMANCE i486 WITH SUPERTUNED MEMORY TECHNOLOGY
- UPGRADES USING FUTURE OVERDRIVE PROCESSORS BASED ON PENTIUM TECHNOLOGY
- ACCELERATED 32-BIT LOCAL BUS GRAPHICS
- BUILT IN ADAPTEC SCSI-II FOR INDUSTRY STANDARD EXPANDABILITY
- INTEGRATED BUSINESS AUDIO FOR WORKGROUP PRODUCTIVITY APPLICATIONS
- THREE SLOTS (INCLUDING TWO EISA SLOTS) FOR HIGH PERFORMANCE EXPANSION
- SNAP TOGETHER CHASSIS & SOFTWARE CONFIGURABILITY FOR EASY SETUP.

THE PROFESSIONAL/GX HAS THE POWER FOR TODAY'S ADVANCED SOFTWARE, AND TOMORROW'S BREAKTHROUGH APPLICATIONS. IT'S SIMPLER, SMALLER, AND MORE POWERFUL THAN ANY OTHER PC, AND IT'S HERE TODAY FROM INTEL.

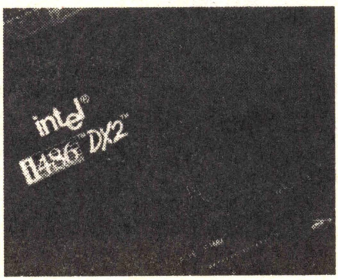
The Intel Professional/GX combines the power of a graphics workstation with the flexibility that PC users expect. The unique SuperTuned memory optimizes the microprocessor performance, and accelerated high resolution local bus graphics make sophisticated software more responsive than ever. On-board SCSI allows for easy external expansion, while integrated audio opens up exciting new capibilities. This innovative combination of performance, integration, and flexibility creates an exciting computer for powerful, productive workgroup computing.

SUPERTUNED FOR SPEED

The Professional/GX is designed to make sure each Intel486 microprocessor will operate at peak performance. That's why the Professional/GX uses SuperTuned memory with an optimized 64-bit design that gives i486™ DX2/66 MHz CPUs a performance boost of up to 25% over many competitive desktop systems.

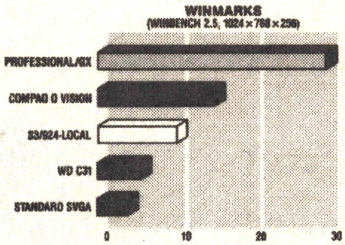
UPGRADABLE BEYOND i486™ PERFORMANCE

The Professional/GX can be upgraded simply by plugging in an Intel OverDrive™ processor—including a future OverDrive processor that will be based on Intel's Pentium™ microprocessor technology. This flexibility means today's high performance Professional/GX will have no problem keeping up with the software developments of tomorrow.

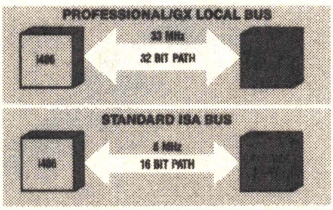


EVEN THE FASTEST i486 DX2/66 CONFIGURATIONS CAN BE EASILY UPGRADED USING INTEL'S NEXT GENERATION OVERDRIVE PROCESSOR

An advanced graphics subsystem gives the Professional/GX amazing graphics responsiveness, increasing drawing and windowing speed to levels far superior to expensive add-in cards.



An advanced *mach32* graphics accelerator, a 32-bit linear frame buffer on the local bus, and up to 2MB of fast VRAM enable this breakthrough performance and eliminate the distractions and delays of scrolling around the workspace and waiting for images to redraw. Users maximize productivity using today's demanding operating systems—like Microsoft Windows 3.1, IBM OS/2 2.0, and SCO ODT—and have the performance headroom for tomorrow's new operating systems like Windows NT.

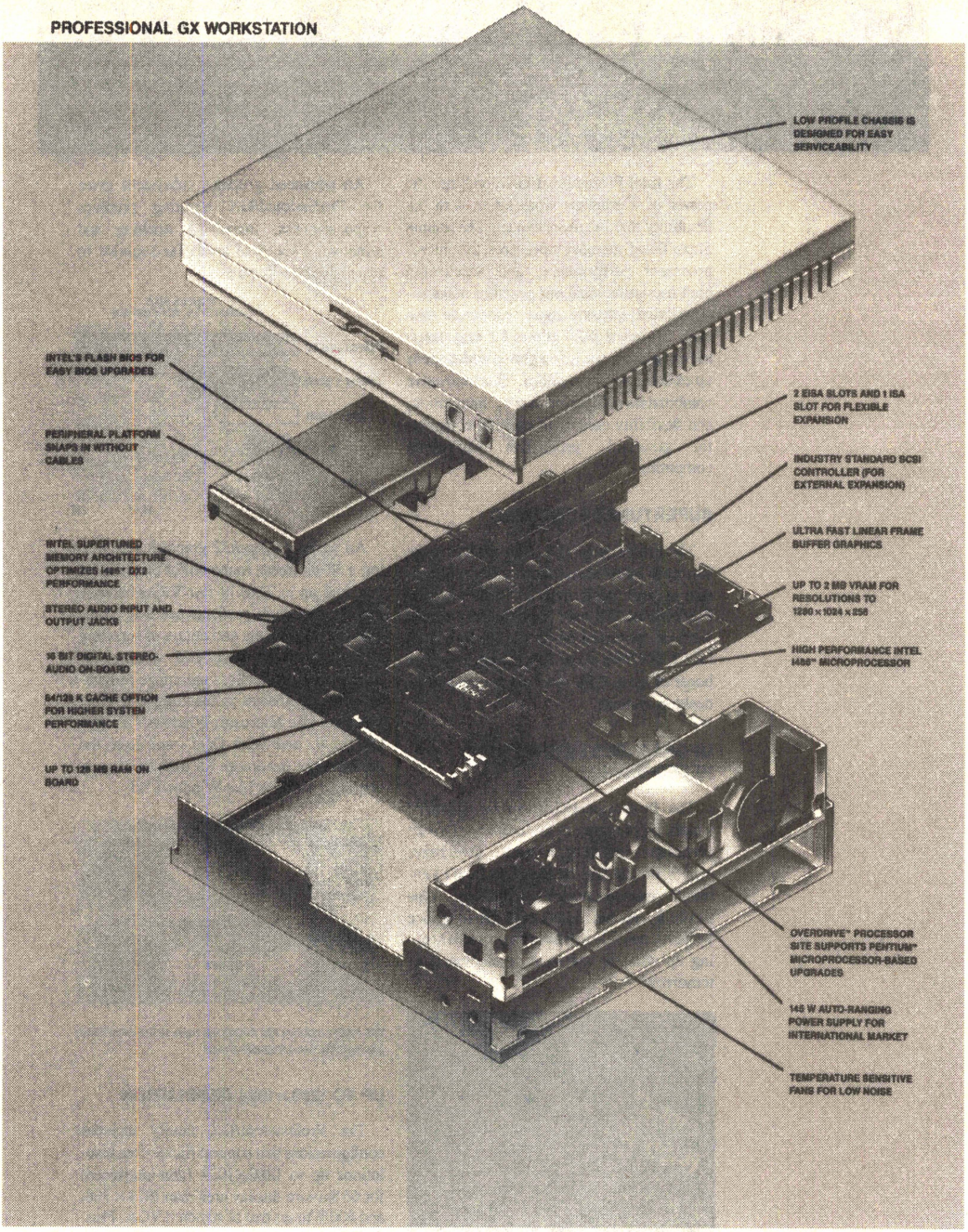


THE PROFESSIONAL/GX PROVIDES 700% MORE GRAPHICS BANDWIDTH THAN STANDARD VGA

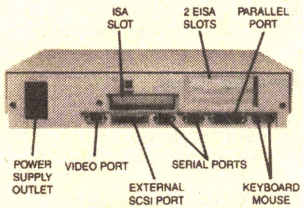
UP TO 1280 x 1024 RESOLUTION

The Professional/GX family includes configurations that support flicker-free resolutions up to 1280 x 1024 (non-interlaced) for 66% more display area than 1024 x 768, and 400% times that of standard VGA. Plus, the Professional/GX can provide 16 bit or 24 bit color depths for up to 16.7 million simultaneous colors and photo-realistic images for advanced imaging, publishing, and video applications.

PROFESSIONAL GX WORKSTATION



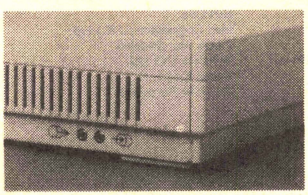
The Professional/GX has Adaptec SCSI integrated on the motherboard for quick connection and configuration of up to seven external devices including hard disks, CD-ROMs, or tape drives. There are three expansion slots (2 EISA, one ISA), 2 peripheral bays, memory expandable to 128MB and cache upgradable to 128KB. All this plus fully integrated I/O provides tremendous flexibility and upgradability.



THE NEW SENSATION: BUSINESS AUDIO

The Professional/GX integrates high quality digital audio on the motherboard allowing you to use Windows 3.1 audio capabilities. Record sounds or speech, and play them back through the internal speaker or through external devices (headphones or speakers) with 16 bit digital stereo and CD quality.

Business audio takes advantage of a new generation of software that lets users embed sounds or voice notes in Microsoft Windows applications, such as spreadsheets, word processing documents, presentations, and electronic mail. And new software enables voice recognition and many other exciting advancements, including full motion video with soundtrack.



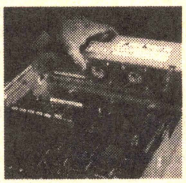
WITH ON-BOARD AUDIO, THE PROFESSIONAL/GX PROVIDES INSTANT ACCESS TO MULTIMEDIA APPLICATIONS.

The Professional/GX is designed to be extremely easy to set up and service, with a unique snap together cable free chassis design.

THE PERIPHERAL PLATFORM SNAPS OUT WITHOUT TOOLS FOR EASY ACCESS TO THE MOTHERBOARD AND DRIVES.



THE POWER SUPPLY MODULE SNAPS OUT EASILY, ALSO WITHOUT TOOLS.



All of the integrated features on the Professional/GX are software configurable, eliminating the need to open the system and change jumpers. Utilities make it simple to configure the system, install software drivers, and update the Intel Flash Memory BIOS. Maximizing memory for network and application software is easy because the BIOS provides a 160KB block of upper memory.

INNOVATIVE ERGONOMIC DESIGN

The Professional/GX is designed as a performance powerhouse in slim, ergonomic packaging. The entire system is only 3″ high with a small footprint but with the strength to support a large 19″ monitor. It is extremely quiet because of two efficient thermally controlled fans.

The Professional/GX is designed for maximum software compatibility for the integrated video and SCSI. Drivers allow many SCSI peripherals to be used in most operating systems.

Optimized video drivers exist for all the major Graphical User Interfaces (Windows 3.1, OS/2 2.0, and SCO ODT) and for popular DOS programs like Autocad. Compatibility with the popular 8514/A standard enables the use of every other major operating system at 1024×768×256 resolution.

The Professional/GX is network certified with Intel's EtherExpress™ and Token-Express™ network cards, guaranteeing broad software support and easy configuration.

OUTSTANDING SERVICE AND SUPPORT

IntelTechDirect™ is a support program designed to bring your product to market faster and keep it running smoothly. A subscription to IntelTechDirect includes factory hotline, electronic bulletin board, and monthly product updates.

The Professional/GX comes with a 12 month warranty with several service options, including mail in to the factory, carry in to a service depot, or "spare by air." Additional service options include on-site service and spare part kits.



BOARD

Processor/Cache	
Processors:	i486 SX microprocessor; 33 MHz i486 DX microprocessor; 33 MHz i486 DX2 microprocessor; 66 MHz
Upgrades:	Intel OverDrive Processor for the i486 SX33/i486 DX33 Intel OverDrive Processor for the i486 DX2/66 (future)
Cache option	Two-way set associative, 64 KB or 128 KB
System Memory	4 SIMM Sockets
Standard Memory	4 MB (two 2MB SIMMs) or 8 MB (two 4 MB SIMMs). (See ordering information.) One open bank for 2 SIMMs shipped on all configurations.
Maximum RAM	128 MB (four 32 MB SIMMs using 8×36 SIMMs)
SIMM Type	512 Kb×36, 1 Mb×36, 2 Mb×36, 4 Mb×36, or 8 Mb×36. Must fill RAM in pairs of SIMMs.
Access time	70 ns or faster
Error Detection	Byte parity
Architecture	64 Bit interleave
Integrated Graphics	
High Resolution Configurations:	1280×1024×256 colors or 1024×768×65,536 colors or 800×600×16.7 million colors, 2 MB VRAM
Standard Resolution Configurations:	1024×768×256 colors, or 640×480×65,536 colors, 1 MB VRAM (upgradable to 1024×768×64,536 colors)
Compatible with	8514/A, VGA, EGA, HGC, MDA
Controller	ATI 68800 on local bus
Video Port	15 pin connector, analog
VESA compliant	8514/A feature connector to off board DAC only
Integrated Audio	
Capabilities	8/16 bit digital mono/stereo, 5.5 KHz to 48 KHz sampling
Input	3.5 mm jack for dynamic microphone or stereo line input
Output	3.5 mm jack for stereo line output to stereo speakers or headphones (also can use PC internal speaker)
Integrated I/O	
2 Serial Ports	Async, RS-232-C, 9 pin
Parallel Port	Centronics, 25 pin, IBM-AT compatible
Mouse Port	PS/2 compatible
Keyboard Port	PS/2 compatible
Floppy Disk Controller	
IDE Hard Disk Connector	
Integrated SCSI	
Controller	Adaptec 6360
Interface	SCSI-2, 50 pin high density connector for external devices only
Expansion Slots	
2 EISA Slots	Full size
1 ISA Slot	Half size

SYSTEM

System Dimensions

Height 3.0″ (7.6 cm) Depth 15.0″ (38.1 cm)
Width 15.5″ (39.4 cm) Weight 20 lbs (9 kg) (no peripherals)

Drive Bays

Floppy bay	3.5″ 1.44 MB floppy drive installed (1″)
Hard Drive Bay	One 3.5″×1″ high IDE drive bay

ELECTRICAL

AC Voltage/Frequency 115/230 V, 50/60 Hz, Auto switching

Available Power (for expansion slots and internal IDE hard drive)

+5V	10.2 Amps	−5V	.4 Amps
+12V	2.6 Amps	−12V	.4 Amps

ENVIRONMENT

Ambient Temp. Operating	+10°C to +40°C
Relative Humidity−Oper.	92% RH at +36°C
Noise	less than 35 dBA @ 1 meter, 25°C

REGULATIONS

Meets or exceeds the following requirements:

Safety	
U.S.	UL 1950
Canada	CSA C22.2 No. 950
Europe	TUV EN 60950 with ZH/618/10.8
EMI/RFI	
U.S.	FCC CFR 47 Part 15, Class B
Canada	DOC CRC c.1374 Class B
Europe	VDE 0871 Class B, CISPR-B

ORDERING INFORMATION

(All systems include 1.44 MB floppy, integrated audio and SCSI.)

High Resolution Graphics (2 MB VRAM, 1280×1024×256 colors or 1024×768×65,536 colors or 800×600×16.7 million colors)

L486D2668FAH	Professional/GX, i486 DX2/66, 8 MB RAM, 64K cache
L486DX338FAH	Professional/GX, i486 DX33, 8 MB RAM

Standard Resolution Graphics (1 MB VRAM, 1024×768×256 colors or 640×480×65,536 colors)

L486D2668FAS	Professional/GX, i486 DX2/66, 8 MB RAM, 64K cache
L486DX334FAS	Professional/GX, i486 DX33, 4 MB RAM
L486SX334FAS	Professional/GX, i486 SX33, 4 MB RAM

Accessory Kits

L486CACHE128	Kit for adding or upgrading to 128KB cache
L486VRAM	Kit for adding 1 MB VRAM to std. res. config.

For more information or the number of your nearest Intel sales office, call 800-548-4725 (U.S. and Canada).

UNITED STATES
Intel Corporation
2200 Mission College Dr.
Santa Clara, CA 95052-8119
800-548-4725

EUROPE
Intel Corporation (U.K.) Ltd.
Pipers Way, Swindon
Wiltshire SN3 1RJ
United Kingdom
Tel: +44 793 696000

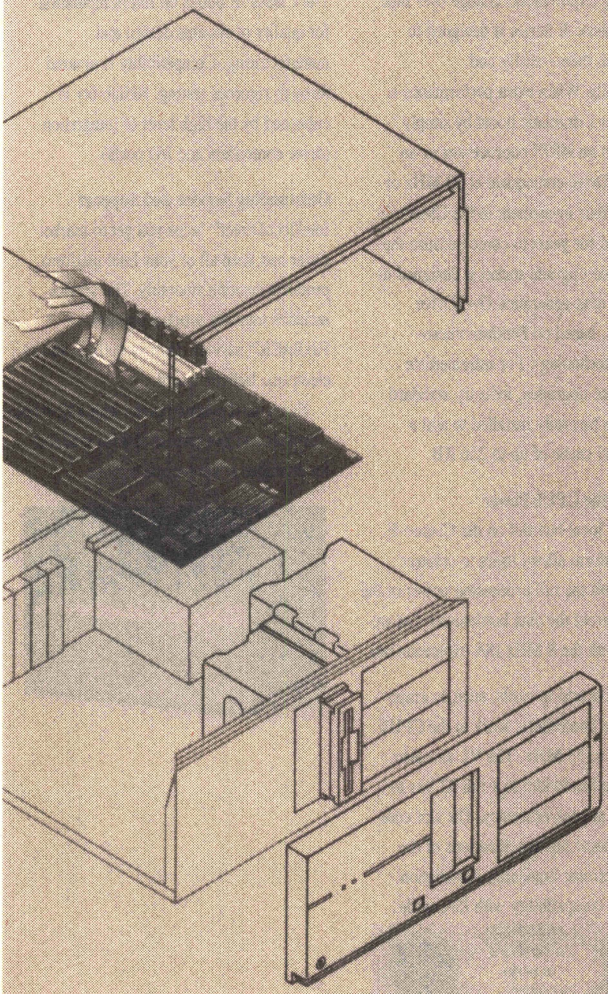
Intel Corporation assumes no responsibility for the use of circuitry other than circuitry embodied in an Intel product. i486, i487, OverDrive, IntelTechDirect, Pentium, EtherExpress and TokenExpress are trademarks of Intel. All other trademarks and product names are the properties of their respective owners. Intel Corporation retains the right to make changes to these specifications at any time without notice.

©INTEL CORPORATION, 1993

ORDER NUMBER: 281555-001

Intel i486™ Classic-E Platforms

DT486SX254F
DT486DX334F
DT486DX2664F



16

intel

October 1993
Order Number: 281565-001

Intel i486™ Classic-E Platforms

A Classic Expandable

Intel's Classic-E Series offers an affordable solution to the OEM or Custom Solution Provider seeking maximum flexibility and expandability in an industry-standard Baby-AT chassis. Besides delivering the needed "slots and watts" for integration, the Classic-E has performance headroom for the future with economical upgrades to OverDrive™ processors that are based on Pentium™ microprocessor technology. All standard I/O features are integrated on the motherboard and one of the seven available expansion slots accepts VESA™-standard local bus add-in cards.

The Standard for Expandability

The Classic-E Series Platform is built with custom configurations in mind. All five peripheral bays provide external access for up to three half-height 5¼" devices and two one-inch high 3½" drives. The seven 16-bit ISA expansion slots are full length. One also accepts a VL-Bus card to allow the user to tap into the full processing speed of the CPU. All of the slots are available because standard I/O features have been integrated on the motherboard, including two serial ports, a parallel port, a keyboard connector, the floppy disk controller and an IDE hard disk controller. Power requirements for the add-in cards and peripherals are never a problem with the ample 200-watt power supply. System memory expands to a total of 64 MB.

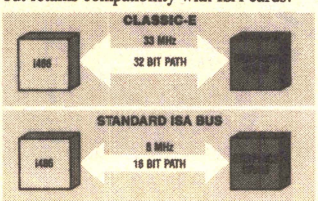
Flexibility for the Future

Computing requirements change over time, and the Classic-E Series is designed to change with them—easily and economically. When extra performance is needed, get a dramatic boost by simply plugging in an i487™ coprocessor or an Intel OverDrive component at 50 MHz or 66 MHz. Your investment in the Classic-E is protected for years to come because the performance upgrade socket is designed to accept the next-generation OverDrive processors, based on Pentium microprocessor technology. For intermediate performance upgrades, industry standard SRAM can be easily installed to add a second level cache of up to 256 KB.

Plug into the CPU's Power

A VESA™ local bus slot on the Classic-E Series Platform allows users to connect directly with the full processing power of the CPU and avoid the data bottlenecks that are inherent with the 8 MHz ISA expansion bus.

I/O intensive add-in cards, such as graphics or drive controllers, work at the CPU's speed—up to 33 MHz. The VL slot also doubles the bandwidth to a full 32 bits to move data at peak efficiency. The slot complies with local bus specifications of the Video Electronic Standards Association, but retains compatibility with ISA cards.



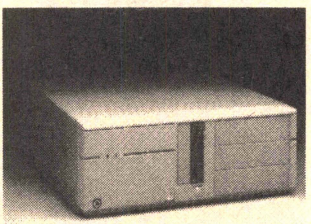
Quality Plus Reliability

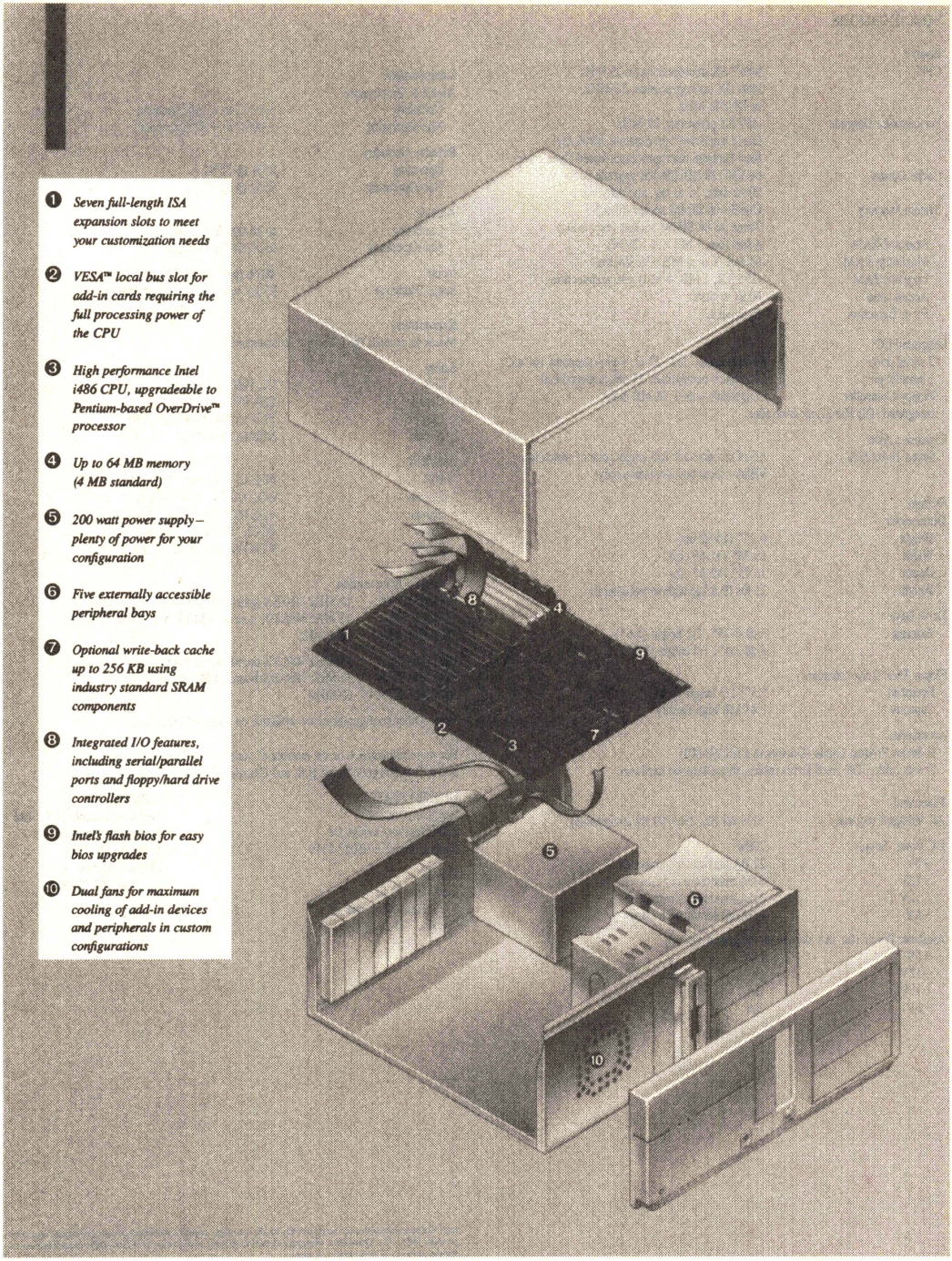
The Classic-E builds on Intel's reputation for quality of systems design and manufacturing. Compatibility is assured through rigorous testing. Reliability is enhanced by the high level of integration (drive controllers and I/O ports).

Outstanding Service and Support

IntelTechDirect™ helps you get to market faster and keep all of your Intel platform products running smoothly. The service includes monthly product updates, FaxBACK™ service, factory hotline, and an electronic bulletin board.

Every Intel system comes with a 12-month warranty and has several options available for extended coverage and service.




1. Seven full-length ISA expansion slots to meet your customization needs
2. VESA™ local bus slot for add-in cards requiring the full processing power of the CPU
3. High performance Intel i486 CPU, upgradeable to Pentium-based OverDrive™ processor
4. Up to 64 MB memory (4 MB standard)
5. 200 watt power supply – plenty of power for your configuration
6. Five externally accessible peripheral bays
7. Optional write-back cache up to 256 KB using industry standard SRAM components
8. Integrated I/O features, including serial/parallel ports and floppy/hard drive controllers
9. Intel's flash bios for easy bios upgrades
10. Dual fans for maximum cooling of add-in devices and peripherals in custom configurations

Specifications

Board

CPU	i486™ SX microprocessor, 25 MHz i486 DX microprocessor, 33 MHz or DX266 MHz
Performance Upgrade	i487 SX processor, 25 MHz Intel OverDrive™ processors; 50/66 MHz Intel Pentium microprocessor-based OverDrive
Cache Option	64 KB/128 KB/256 KB upgrade option 20 ns data, 15 ns tag, and dirty bit
System Memory	One 36-bit SIMM socket (filled) Three 36-bit SIMM sockets (expansion)
Standard RAM	4 MB (one 1 MB×36 SIMM)
Maximum RAM	64 MB (four 4 MB×36 SIMMs)
Upgrade RAM	(256 KB, 1 MB, 4 MB)×36 architecture
Access time	80 ns or faster
Error Detection	Byte parity

Integrated I/O

2 serial ports	Asynchronous, RS-232-C, 9-pin (supports 16C452)
1 parallel port	Centronics compatible, 25-pin, bidirectional
Floppy Controller	Compatible with 2.88 MB disks

Integrated IDE Hard Disk controller

Expansion Slots

Seven 16-bit ISA	All of the slots are full-length, one of which is a VESA local bus slot (slave only)

System

Dimensions

Height:	6.25″ (15.92 cm)
Width:	16.35″ (41.65 cm)
Depth:	16.75″ (42.67 cm)
Weight:	21 lbs (9.5 kg) without peripherals

Drive Bays

External	3 @ 5.25″; 1/2 height (1.6″) 2 @ 3.5″; 1/3 height (1.0″)

Floppy Disk Drive (standard)

Footprint	3.5″; 1/3 height (1.0″)
Capacity	1.44 MB high density

Accessories

E-Series Product Guide (Document # 612169-001)
Power cable, IDE Hard Drive cables, miscellaneous hardware

Electrical

AC Voltage/Frequency	115v/60 Hz; 230v/50 Hz (switchable)
DC Power Supply	200w
+5V	22.0A maximum continuous
+12V	8.0A maximum continuous
−12V	0.5A maximum continuous
−5V	0.5A maximum continuous

Available Power (for ISA slots in an integrated system)

+5V	14A
+12V	4A
−12V	.3A
−5V	.3A

Environment

Ambient Temperature

Operating	+10°C to +35°C ambient
Non-Operating	−40°C to +70°C ambient

Relative Humidity

Operating	85% @ 35°C
Non-Operating	95% @ 35°C

Altitude

Operating	to 10,000 feet (3,048 m)
Non-Operating	to 50,000 feet (15,240 m)

Noise	40 db (peripherals idle, at 1 meter)
Static Discharge	25 kV with no component damage

Regulations

Meets or exceeds the following requirements:

Safety

U.S.	UL 1950, 1st edition
Canada	CSA 950
Europe	TUV to EN 60950, IEC 950
Nordic	NEMKO EN 60950

EMI/RFI

U.S.	FCC CFR 47 Part 15, Level B
Canada	DOC CRC c. 1374 Class B
Europe	VDE 0871 Class B, CISPR-22 Class B
Japan	VCCI, CISPR-22 Class B
Nordic	NEMKO, EN 55 022, Class B

Ordering Information

DT486SX254F	25 MHz i486 SX system; 4 MB RAM, floppy drive
DT486DX334F	33 MHz i486 DX system; 4 MB RAM, floppy drive
DT486DX2664F	2/66 MHz
BDT486SX254	25 MHz i486 SX board, 4 MB RAM
BDT486DX334	33 MHz i486 DX board, 4 MB RAM
BDT486DX2664	66 MHz

Note: Other configurations are available on a special order basis

For more information or the number of your nearest Intel sales office, call 800-628-8686 (good in the U.S. and Canada).

UNITED STATES
Intel Corporation
2200 Mission College Dr.
Santa Clara, CA 95052-8119

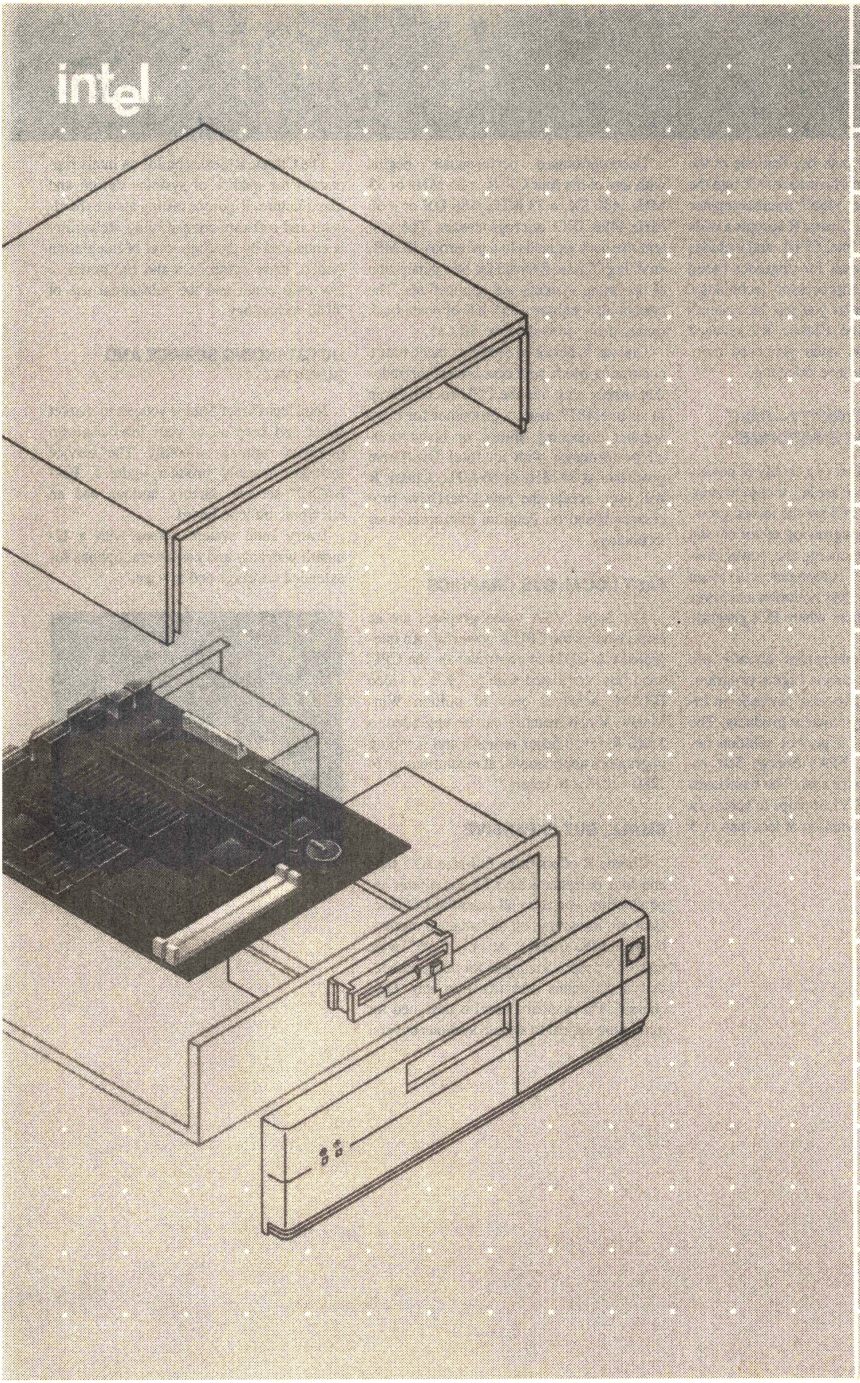
JAPAN
Intel Japan K.K.
5-6 Tokodai, Tsukuba-shi
Ibaraki, 300-26

EUROPE
Intel Corporation (U.K.) Ltd.
Pipers Way, Swindon
Wiltshire SN3 1RJ
United Kingdom
Tel: +44 793 696000

Intel Corporation assumes no responsibility for the use of any circuitry, other than circuitry embodied in an Intel product. i486, i487, OverDrive, Pentium and IntelTechDirect are trademarks of Intel. Other brands and names are the property of their respective owners.

© INTEL CORPORATION 1993

ORDER NUMBER 281565-001



INTEL i486™ CLASSIC-R PLATFORMS

R486SX254F

R486SX334F

R486DX334F

R486DX2664F

281571-1

16

October 1993
Order Number: 281571-001

Intel's Classic R puts key features of the next generation of performance PCs into the hands of entry-level i486™ microcomputer desktop customers. Classic R accepts a wide range of the fastest Intel CPUs, and includes an OverDrive™ socket for upgrades based on Pentium™ microprocessor technology. Local bus graphics let you see the system's responsiveness. And Classic R's reduced power consumption saves you even more than its cost-competitive price tag.

USE LESS ELECTRICITY—SAVE MONEY AND THE ENVIRONMENT

Classic R is the first of a family of power-efficient products that are key to the environmental revolution. PCs are the fastest growing load in the fastest growing sector of total energy demand. Reducing the power consumption of office equipment will mean fewer power plants, less pollution and lower cooling costs in offices where PCs generate a lot of heat.

Leading edge companies already are adopting the EPA Green Lights program, which proposes a two-year payback on investments in energy-efficient products. The Classic R-Series is a perfect solution because it meets the EPA's Energy Star requirements at no extra cost. The baseboard uses several CMOS VLSI chips to achieve a typical power consumption of less than 11.5 watts when idle.

Uncompromised performance begins with any of the four CPUs: a 25 MHz or 33 MHz i486 SX, a 33 MHz i486 DX or a 66 MHz i486 DX2 microprocessor. The system memory expands to a generous 32 MB, enabling Classic R to run the new generation of operating systems and applications. The system also supports 256 KB of write-back cache, using cost-effective SRAM.

Classic R-Series is ready to meet future computing needs with economical upgrades that simply plug into the CPU socket. Swap in an Intel487™ math coprocessor for more number crunching ability, or boost overall performance with an Intel OverDrive processor at 50 MHz or 66 MHz. Classic R will even accept the new OverDrive processors based on Pentium microprocessor technology.

FAST LOCAL BUS GRAPHICS

The Super VGA video graphics are as responsive as the CPU is powerful. An integrated CL-GD5424 controller on the CPU local bus, combined with 512 KB of video DRAM, achieves over 10 million WinMarks. Video memory can be upgraded to 1 MB for even faster redraws and scrolling in graphics applications, at resolutions up to 1280×1024×16 colors.

SMALL, BUT EXPANSIVE

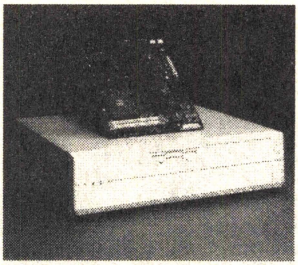
Classic R offers three full-size AT* slots and four peripheral bays for maximum expandability—because all essential I/O functions are integrated on the smallest motherboard on the market. At only 7.95″ wide and 9.4″ deep, it's suitable for many embedded applications and fits into any LPX chassis. The standard bezel is designed for maximum appeal to style-conscious buyers.

The Classic R-Series builds on Intel's reputation for quality of systems design and manufacture. Rigorous testing assures hardware and software compatibility. Reliability is enhanced by the high level of integration (video, drive controllers and I/O ports), a low chip count and the substantial use of VLSI technology.

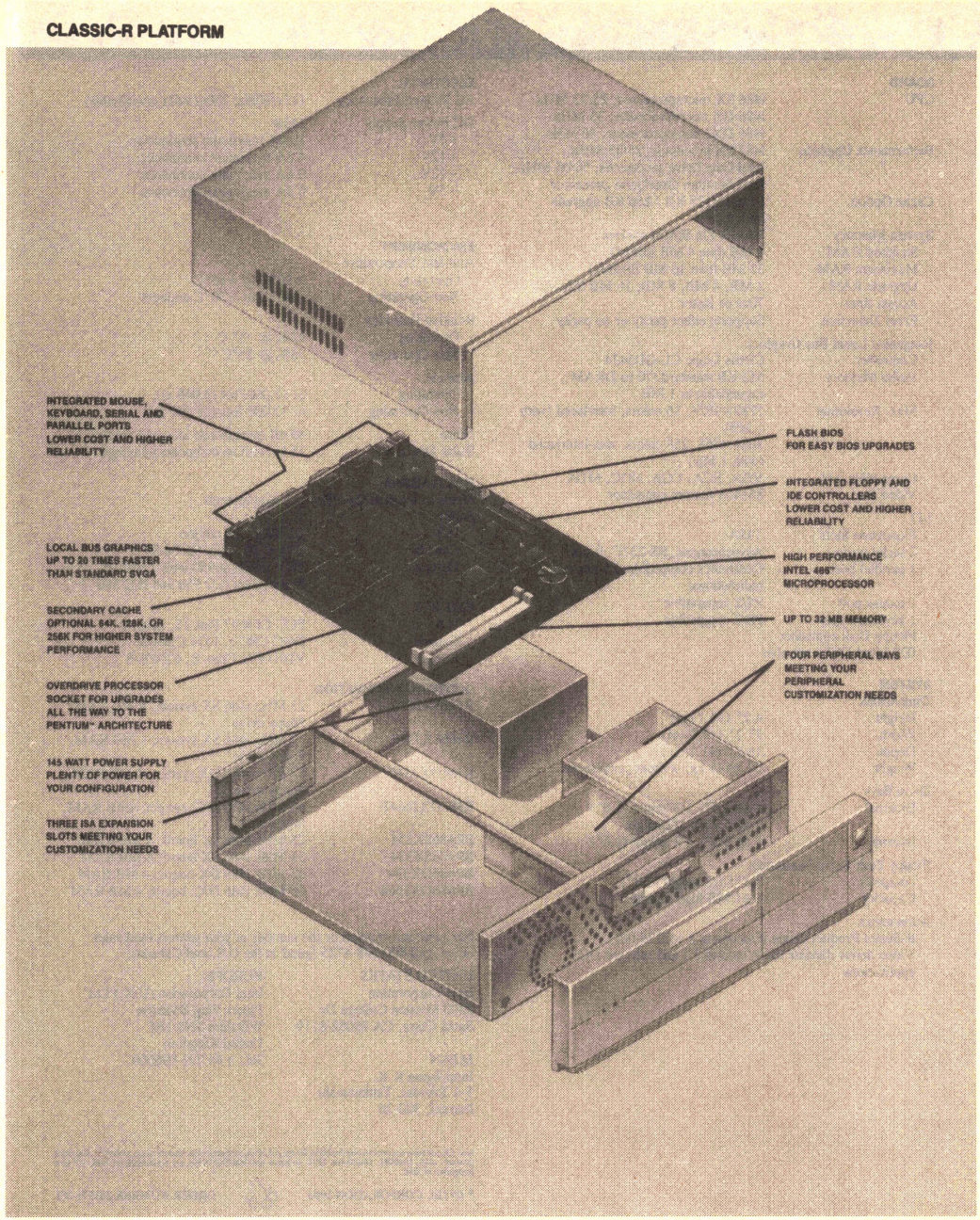
OUTSTANDING SERVICE AND SUPPORT

IntelTechDirect™ helps you get to market faster and keep all of your Intel platform products running smoothly. The service includes monthly product updates, FaxBACK™ service, factory hotline, and an electronic bulletin board.

Every Intel system comes with a 12-month warranty and has several options for extended coverage and service.



CLASSIC-R PLATFORM



BOARD

CPU	i486 SX microprocessor; 25/33 MHz i486 DX microprocessor; 33 MHz i486 DX2 microprocessor; 66 MHz
Performance Upgrade	i487 SX processor; 25/33 MHz Intel OverDrive processors; 50/66 MHz Intel Pentium OverDrive processor
Cache Option	64 KB / 128 KB / 256 KB upgrade option
System Memory	Two 72-Pin SIMM sockets
Standard RAM	4 MB (one 4 MB SIMM)
Maximum RAM	32 MB (two 16 MB SIMMs)
Upgrade RAM	2 MB, 4 MB, 8 MB, 16 MB SIMMs
Access time	70ns or faster
Error Detection	Supports either parity or no parity
Integrated Local Bus Graphics	
Controller	Cirrus Logic CL-GD5424
Video Memory	512 KB standard (70 ns DRAM), expandable to 1 MB
Max. Resolution	1280×1024, 16 colors, interlaced (with 1 MB) 1024×768, 256 colors, non-interlaced (with 1 MB)
Compatible with	VGA, EGA, CGA, HGC, MDA
Video Port	8514/A feature connector
I/O	
Expansion Slots	3 ISA
2 serial ports	Asynchronous, RS-232C, 9-pin
1 parallel port	Centronics compatible, 25-pin, bidirectional
1 mouse port	PS/2 compatible
1 keyboard port	PS/2 compatible
Floppy Disk controller	
IDE Hard Disk controller	

SYSTEM

Dimensions	
Height	4.2″ (10.7 cm)
Width	17.2″ (43.7 cm)
Depth	16.2″ (41.1 cm)
Weight	20 lbs (9 kg) without peripherals
Drive Bays	
External	1 @ 3.5″; 1/3 height (1.0″) 2 @ 5.25″; 1/2 height
Internal	1 @ 3.5″; 1/2 height (1.6″)
Floppy Disk Drive (standard)	
Footprint	3.5″; 1/3 height (1.0″)
Capacity	1.44 MB high density

Accessories

R-Series Product Guide (Document # 612775-001)
Video driver diskette for Windows 3.1 and selected DOS applications

ELECTRICAL

AC Voltage/Frequency	115v/60Hz; 230v/50Hz (switchable)
DC Power Supply	145w
+5V	18.0A maximum continuous
+12V	4.2A maximum continuous
−12V	0.6A maximum continuous
−5V	0.3A maximum continuous

ENVIRONMENT

Ambient Temperature	
Operating	+10°C to +35°C
Non-Operating	−40°C to +70°C amibent
Relative Humidity	
Operating	85% @ 35° C
Non-Operating	95% @ 35°C
Altitude	
Operating	to 10,000 feet (3,048 m)
Non-Operating	to 50,000 feet (15,240 m)
Noise	40 dB (peripherals idle, at 1 meter)
Static Discharge	25 kv with no component damage

REGULATIONS

Meets or exceeds the following requirements.

Safety	
U.S.	UL 1950, 1st edition
Canada	CSA C 22.2 No. 950M-89
Europe	TUV to EN 60950 with ZHI/618 Ergonomics IEC 950 and IEC 380
EMI/RFI	
U.S.	FCC CFR 47 Part 15, Level B
Canada	DOC CRC c, 1374 Class B
Europe	VDE 0871 Class B, CISPR-B

ORDERING INFORMATION

R486SX254F	25 MHz i486 SX system; 4 MB RAM, floppy drive
R486SX334F	33 MHz i486 SX system; 4 MB RAM, floppy drive
R486DX334F	33 MHz i486 DX system; 4 MB RAM, floppy drive
R486DX2664F	66 MHz i486 DX2 system; 4MB RAM, floppy drive
BR486SX254	25 MHz i486 SX board; 4 MB RAM
BR486SX334	33 MHz i486 SX board; 4 MB RAM
BR486DX334	33 MHz i486 DX board; 4 MB RAM
BR486DX2664	66 MHz i486 DX2 board; 4 MB RAM

For more information or the number of your nearest Intel sales office, call 800-548-4725 (good in the U.S. and Canada).

UNITED STATES
Intel Corporation
2200 Mission College Dr.
Santa Clara, CA 95052-8119

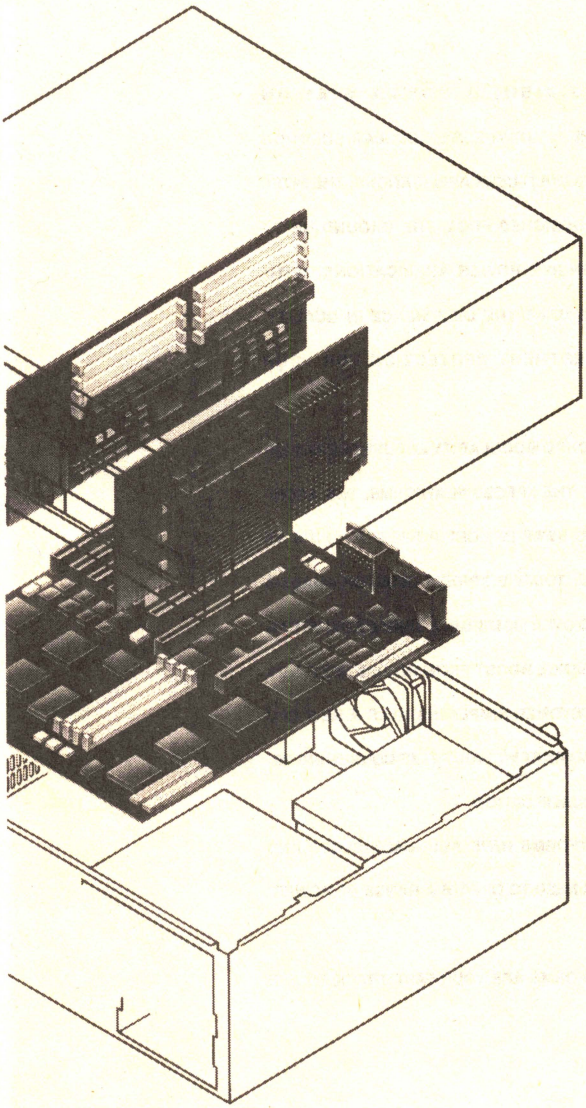
EUROPE
Intel Corporation (U.K.) Ltd.
Pipers Way, Swindon
Wiltshire SN3 1RJ
United Kingdom
Tel: +44 793 696000

JAPAN
Intel Japan K.K.
5-6 Tokodai, Tsukuba-shi
Ibaraki, 300-26

Intel Corporation assumes no responsibility for the use of any circuitry, other than circuitry embodied in an Intel product. i486, Intel487, OverDrive, DVI, Pentium and IntelTechDirect are trademarks of Intel. AT is a trademark of IBM.

© INTEL CORPORATION 1993

ORDER NUMBER 281571-001



XPRESS DESKTOP PLATFORMS

i486™ AND PENTIUM™ MICROPROCESSORS

16

October 1993
Order Number: 281377-003

FROM START TO FINISH, A POWER STATION

XPRESS REPRESENTS THE LEADING EDGE MODULAR SOLUTION FOR POWER DESKTOP AND MULTI-USER APPLICATIONS. THE INTEL XPRESS DESKTOP WAS DESIGNED FROM THE GROUND UP TO TACKLE COMPUTE-INTENSIVE POWER APPLICATIONS, WITH UNIQUE FEATURES THAT MAKE THE DIFFERENCE IN QUALITY, RELIABILITY AND INVESTMENT PROTECTION FOR YOUR CUSTOMERS.

AN INGENIOUS APPROACH TO MODULARITY ENSURES PEAK PERFORMANCE FROM ALL OF THE XPRESS PLATFORMS. THE ARCHITECTURE ALLOWS YOU TO SWAP IN MORE POWERFUL CPU'S AS THEY BECOME AVAILABLE. TODAY'S XPRESS DESKTOP CHASSIS WILL TAKE YOU FROM 20 TO OVER 100 MIPS IN A SINGLE BOX! WITH A GUARANTEE OF PERFORMANCE BOOST FROM INTEL'S NEXT GENERATION CPU'S! YOUR CUSTOMER SIMPLY INSTALLS A NEW CPU MODULE AND SPEEDS TO MARKET WITHOUT TIME-CONSUMING RECONFIGURATION AND REQUALIFICATION.

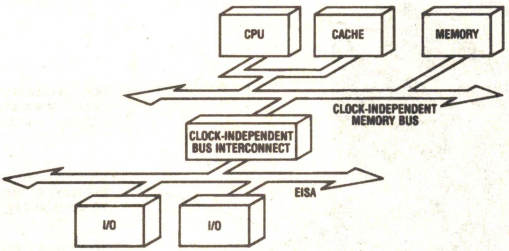
XPRESS DESKTOP PLATFORMS HAVE ABUNDANT SLOTS AND WATTS—ALL YOU'LL EVER NEED TO CREATE A RANGE OF POWERFUL APPLICATIONS.

XPRESS DESKTOP PLATFORMS ARE YOUR FAST TRACK TO THE FUTURE.

MODULAR AND SCALABLE: PENTIUM™ PROCESSORS AND BEYOND

Xpress Deskside/MX platforms provide midrange server class performance with a selection of CPUs, including blazing speeds of over 100 MIPs using today's Pentium™ microprocessors. Our servers are designed to provide peak performance for years to come at even faster speeds using next-generation processors and I/O technology.

The latest Intel microprocessors can be used instantly in Xpress platforms. The unique modular architecture isolates the fastest evolving technology – the CPU and cache – on a replaceable card. This benefits OEMs and custom solution providers because it simplifies product design, reduces sales development requirements, and provides customers with a way to economically upgrade their systems.



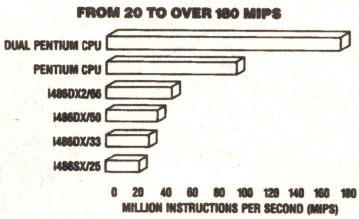
DESIGNED FOR THE FUTURE

Xpress platforms are not only modular, they're scalable, accommodating both today's processors and tomorrow's as well. The 64-bit interleaved memory architecture provides exceptionally high transfer speeds (133 Mbytes per second peak transfer). Since the CPU-memory access is clock-independent, Xpress is able to support tomorrow's CPUs of up to 100 MHz *at full performance*, and as a plus, the memory architecture supports industry standard fast page 1 MB through 32 MB SIMMs for maximum flexibility and affordability.

With a range of single processor modules, and dual Pentium microprocessor modules on the horizon, Xpress/MX provides super-server class performance of over 180 MIPS! The result is a world-class server platform which allows our customers to demonstrate award winning server performance today, and which can provide peak performance for years to come.

POWER FROM 20 TO OVER 180 MIPS

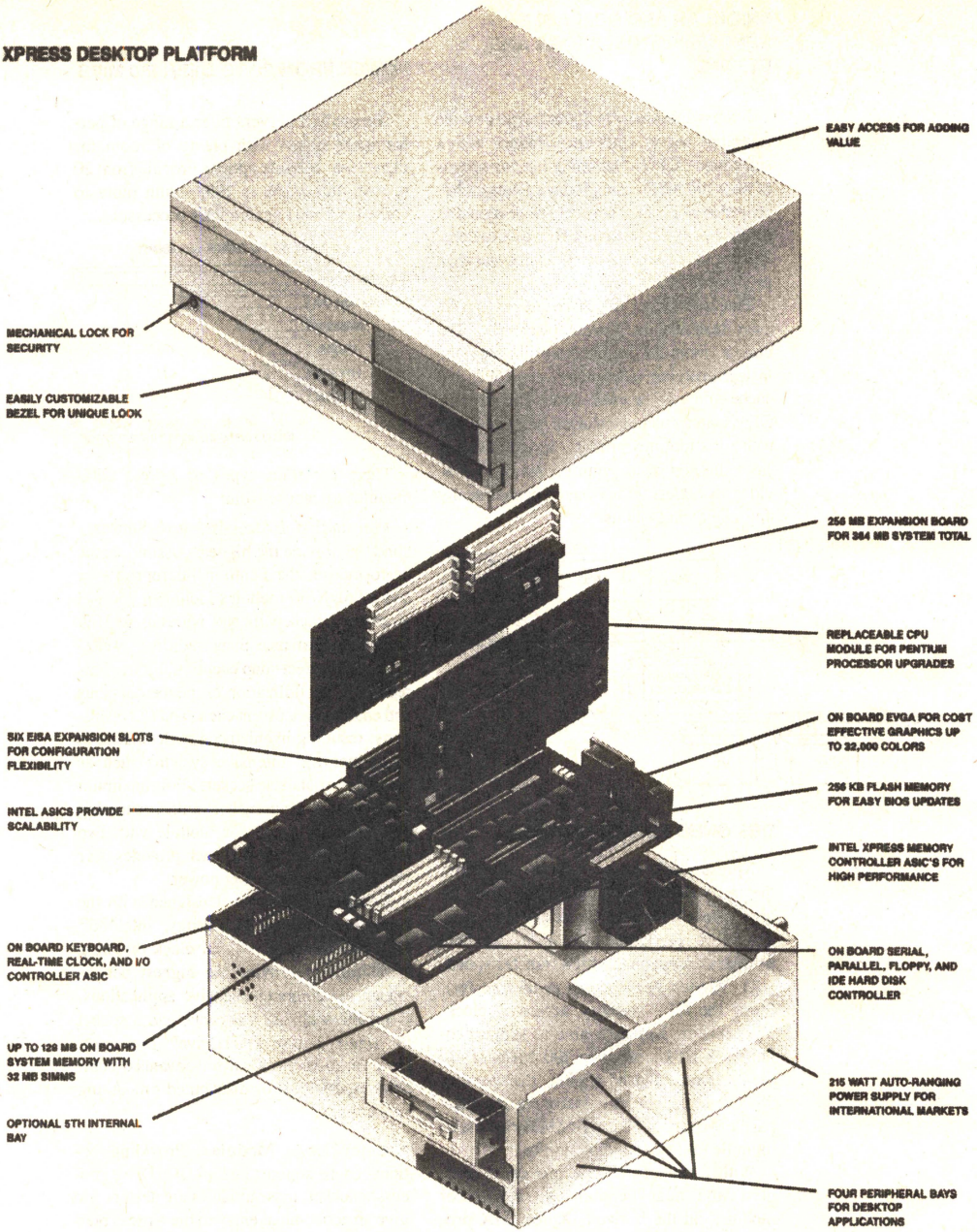
Xpress/MX servers offer a range of performance points with plenty of room for expansion. Module options provide from 20 to over 180 MIPS in 1993 – with more to come from next generation's processors.



There are three types of Xpress CPU modules to choose from:

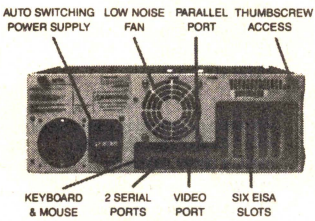
- **Transaction Processing and Server.** Tuned to provide the highest possible server performance, the Pentium microprocessor and i486DX/50 modules include a 256 KB write-back cache which provides up to 36% higher performance than traditional write-through or direct-map cache solutions. This reduces CPU utilization of the memory bus and enables concurrent cache and I/O operations, resulting in enhanced performance in multi-user I/O intensive systems such as network or database servers. For maximum performance, a symmetric multi-processing dual pentium processor module with two level 2 caches of 256 KB each provides over 180 MIPS of processing power.
- **Compute Intensive.** Designed with the latest Intel silicon technology, Intel486™ DX2 modules use internal clock doubling techniques to provide the highest performance for compute-intensive applications. Complete with 128 KB cache and a socket for Intel's Future OverDrive™ processor, these modules are ideal for advanced computer simulations or distributed processing applications.
- **Entry Level Modules.** Providing optional cache support and an OverDrive processor socket, these modules are designed to give an economical entry to the Xpress platforms and are ideal for file servers or customers upgrading their existing i386™ architecture designs.

XPRESS DESKTOP PLATFORM

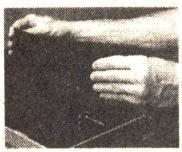


PLENTY OF ROOM FOR EXPANSION

With integrated SVGA windows accelerated graphics, an IDE DMA hard disk interface, a floppy disk controller and essential I/O functions, Xpress platforms eliminate the need for three additional I/O add-in boards typically found in desktop PCs. This high level of integration reduces customization and service costs and frees EISA expansion slots for adding other capabilities.



We've paid particular attention to ease of configuration and service by making the chassis extremely simple to open and add value. The use of Flash BIOS radically simplifies BIOS upgrades. You can upgrade the system BIOS through the onboard floppy device or a communications link.



TO UPGRADE AN XPRESS PLATFORM TO A PENTIUM PROCESSOR, SIMPLY INSTALL THE XPRESS MODULE FOR 64 BIT PERFORMANCE!

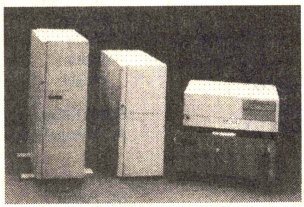
Xpress platforms provide six EISA bus master expansion slots, five peripheral bays (two 5.25″ and three 3.5″) and memory expansion through on-board SIMM modules and a plug-in, parity memory module. Memory can be expanded to 384 MB using industry-standard SIMM modules.

COMPATIBILITY, SERVICE AND SUPPORT YOU CAN DEPEND ON

Intel Xpress platforms are designed for complete compatibility with all popular operating systems, application software packages and peripheral devices. They also comply with Novell's Netware* networking operating systems, Banyan*, Microsoft* LAN Manager, DEC PATHWORKS,* Indeo™ video technology, and for Digital Video Interactive (DVI®) multimedia applications.

FAMILY OF CHASSIS

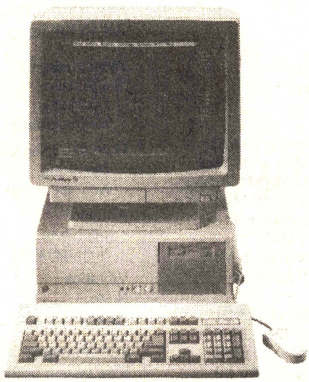
To provide complete flexibility, Xpress platforms are available in a range of different form factors—each utilizing the same scalable baseboard technology. In addition to the desktop, Xpress is available in a mainstream deskside server, a premium server tower, a rack mountable chassis, or as a board level solution.



OUTSTANDING SERVICE AND SUPPORT

Our support services are geared to the unique needs of OEMs and custom solution providers. IntelTechDirect™ helps you get to market faster and keep all of your Intel platform products running smoothly. The service includes monthly product updates, FaxBACK™ service, factory hotline, and an electronic bulletin board.

Every Intel system comes with a 12-month warranty and has several options for extended coverage and service.



XPRESS DESKTOP PLATFORMS ARE ROBUST FOUNDATIONS FOR HIGH-PERFORMANCE CAD, CAE, FINANCIAL ANALYSIS AND SERVER APPLICATIONS.

QUALITY YOU CAN SEE.

Xpress platforms feature a number of quality extras to extend reliability. Metal-tab SIMM connectors reduce memory failures due to socket cracking. The on-board floppy and hard disk connectors are shrouded and locking to provide a good, solid connection. All our systems have transient voltage suppressors added to the serial port lines to absorb damaging voltage spikes. And a comprehensive testing program certifies every Xpress system to the demanding North American FCC Class B, Canadian DOC and European VDE-B specifications, as well as to UL, CSA and IEC 950 standards.

ORDERING INFORMATION:

Base Platform	CPU module shipped separately
XBASE6E4F-B	4 MB Desktop
XBASE6E16F-B	16 MB Desktop
Processor Module	
BXCPUPENT60	60 MHz Pentium Processor w/ 256 KB cache
BXCPU48650	50 MHz i486DX w/ 256 KB cache
BXCPU486DX266-B	66 MHz i486DX2 w/ 128 KB cache
BXCPU48633-B	33 MHz i486 DX module
BXCPU486SX25-B	25 MHz i486SX module
Accessories	
BXMEM0	Memory expansion card up to 256 MB
XBOTFLPKIT	Kit to put top floppy drive into the bottom bay
XDIAGNOSTICS	Complete system diagnostics
XDT5THBAY	Additional internal 1″ drive bracket for fifth drive
XDTSTAND	Vertical stand
Baseboard	CPU module shipped separately
BXBASE6E0-B	0 MB baseboard

Additional configurations and processor modules available. For more information or the number of your nearest Intel sales office, call the numbers below:

UNITED STATES
Intel Corporation
2200 Mission College Dr
Santa Clara, CA
95052-8119
Tel: 800-548-4725

EUROPE
Intel Corporation (UK) Ltd
Pipers Way, Swindon
Wiltshire SN3 1RJ
United Kingdom
Tel: +44 793 696000

JAPAN
Intel Japan K.K.
5-6 Tokodai, Tsukuba-shi
Ibaraki, 300-26
Tel: +81-29847-8522

SPECIFICATIONS

SYSTEM BASEBOARD

System Baseboard Memory:	
Standard RAM	4 MB (512 KB×36 fast page mode double sided (SIMMs)
Maximum RAM	128 MB (8 MB×36 fast page mode double sided SIMMs)
Architecture:	64 bit, interleaved
Peak Transfer bandwidth:	133 MB/s
Effective Transfer bandwidth	82 MB/s
Access Time	80ns (or faster)
Error Detection	Byte Parity
Total memory:	
Expansion module	256 MB with 32 Mbit DRAMS
System baseboard	128 MB with 32 Mbit DRAMS
Total	384 MB with 32 Mbit DRAMS
I/O Expansion Slots:	
Six 32-bit EISA:	All bus master slots
Integrated I/O:	
2 serial ports	Async, RS-232-C, 25-pin
1 parallel port	Centronics and PS/2-compatible, 25-pin
1 mouse port	PS/2-compatible
1 keyboard port	PS/2-compatible
Integrated Fixed Disk	
Controller Interface:	IDE (2 drives), DMA transfers
Integrated Graphics	
Controller:	Western WD90C31 with Windows Acceleration
Video Memory	512 KB (60 ns DRAM) (sockets for easy upgrade to 1 MB DRAM)
Maximum Resolutions	800×600, 32,000 colors 1024×768, 256 colors 1280×1024, 16 colors
Compatible with:	MDA, CGA, HGA, EGA, VGA, SVGA, and EVGA
Output ports:	15-pin, analog Internal 8514/A Video feature connector (VESA-compatible)
Board Dimensions:	Width 10.45″ Depth 13.00″
CPU Module:	Depth 12.08″ (30.7 cm)
(Some modules vary)	Height 4.48″ (11.4 cm)

CPU SPECIFICATIONS, MODULES (choice of):

Pentium Processor	Pentium microprocessors at 60 MHz
Modules:	82495 DX; zero wait state write-back mode with 256 KB SRAM (82490 DX)
i486 DX/50 Module:	i486DX microprocessor at 50 MHz
Cache:	82495 DX; write-back mode with 256 KB SRAM (82490 DX)
i486 DX2/66 Module:	i486 DX2 microprocessor at 66 MHz
Cache:	8 KB internal + 128 KB L2 write-through
i486 DX/33 Module:	i486 DX microprocessor at 33 MHz
Cache:	8 KB internal + optional 128 KB L2
i486 SX/25 Module:	i486 SX microprocessor at 25 MHz
Cache:	8 KB internal + optional 128 KB L2

MEMORY EXPANSION MODULE

Model BXMEM0:	
Standard RAM	0 MB
Maximum RAM	256 MB (8 Mb×36) SIMMs
Access Time	80 ns/70 ns
Error Detection	Byte Parity

©Intel Corporation 1993

Order Number: 281377-003

SYSTEM SPECIFICATIONS:

Floppy Disk	
Format	3.5″, half-height
Capacity	1.44 MB
Drive Bays	
External	Two 5.25″ width, half-height
External	One 3.5″ width, half-height
Internal	Two 3.5″ width, half-height (one is optional)
System Dimensions:	
Height	6.2″ (15.8 cm)
Width	17.3″ (43.9 cm)
Depth	17.1″ (43.4 cm)
Weight	38 lbs (17.2 kg)

ELECTRICAL

AC Voltage/Frequency	Autoranging switching power supply, 100–120/200–240, 60/50 Hz monitor power connector
DC Power	215 W
+5V	32 A maximum continuous
+12V	4.0 A maximum continuous
−12V	0.5 A maximum continuous
−5V	0.2 A maximum continuous

REGULATIONS

Meets or exceeds:

Safety:
UL 1950, 1st edition
CSA C22.2 No. 950-89
TUV Bauart to EN 950/IEC950/ZH1618
NEMKO EN60 950

EMI:
FCC Class B CFR #47 Parts 2 and 15, Certified Class B
VDE 087/6.78 Level B (Vfg 1046)
CISPR 22/85 Class B
VCCI Class 2 ITE
DOC CRC c.1374 Class B
EN 55 101 Series

ENVIRONMENT

Temperature:	
Operating	+10°C to +35°C ambient
Non-operating	−40°C to +65°C ambient
Humidity (Non-condensing wet-bulb max 29°C)	
Operating	20% to 80% @ 35°C
Non-operating	20% to 92% @ 65°C
Altitude:	
Operating	10,000 ft. (3048 m)
Non-operating	50,000 ft (15,240 m)
Vibration:	Random variation 0 Hz–500 Hz
Noise:	40 db @ 23°C (peripherals idle, at 1 m)
Static discharge:	Tested to 25 kV; no component damage

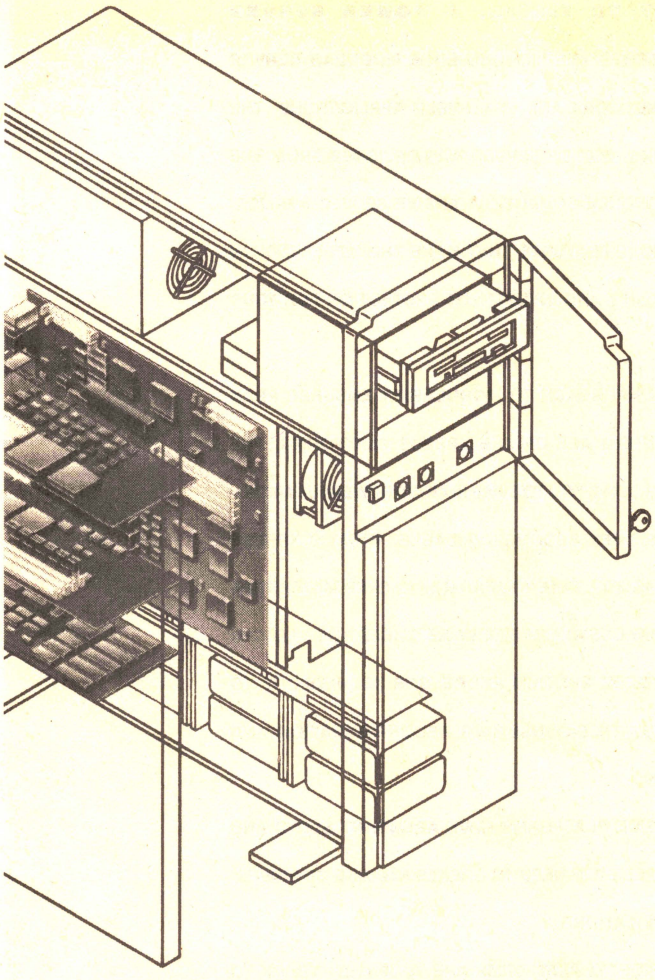
ACCESSORIES:

Base system includes X-Series Desktop Product Guide, EISA configuration utility software, video software drivers, drive rails for additional hard drives, power cable, floppy and hard drive cables.

NETWORK/OPERATING SYSTEM COMPLIANCE:

Banyan, DEC PATHWORKS for DOS, IBM OS/2, LAN Manager, Microsoft Windows, Windows NT, Novell Netware, SCO UNIX, Univel

i486, Intel486, Pentium, DVI, i386, IntelTechDirect and OverDrive are trademarks of Intel Corporation. Banyan is a trademark of Banyan Systems Inc., OS/2 is a trademark of IBM, Netware is a trademark of Novell, SCO is a trademark of Santa Cruz Operations, UNIX is a trademark of AT&T, Windows is a trademark of Microsoft Corporation, DEC Pathworks is a trademark of Digital Equipment Corporation.



INTEL XPRESS/MX SERVER PLATFORMS

i486™ AND PENTIUM™ MICROPROCESSORS

16

October 1993
Order Number: 281379-004

FROM START TO FINISH, A POWER SERVER

XPRESS REPRESENTS THE LEADING EDGE MODULAR SERVER SOLUTION FOR NETWORK AND MULTI-USER APPLICATIONS. THE INTEL XPRESS/MX DESKSIDE SERVER WAS DESIGNED FROM THE GROUND UP TO TACKLE COMPUTE-INTENSIVE SERVER APPLICATIONS, WITH UNIQUE FEATURES THAT MAKE THE DIFFERENCE IN QUALITY, RELIABILITY AND INVESTMENT PROTECTION FOR YOUR CUSTOMERS.

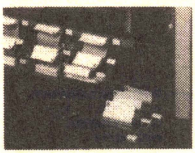
AN INGENIOUS APPROACH TO MODULARITY ENSURES PEAK PERFORMANCE FROM ALL OF THE XPRESS PLATFORMS. THE ARCHITECTURE ALLOWS YOU TO SWAP IN MORE POWERFUL MICROPROCESSORS AS THEY BECOME AVAILABLE. TODAY'S XPRESS DESKSIDE CHASSIS WILL TAKE YOU FROM 20 TO OVER 180 MIPS IN A SINGLE BOX! YOUR CUSTOMER SIMPLY REPLACES THE OLD CPU MODULE AND SPEEDS PENTIUM PROCESSOR TECHNOLOGY TO MARKET WITHOUT TIME-CONSUMING RECONFIGURATION AND REQUALIFICATION.

XPRESS DESKSIDE PLATFORMS HAVE ABUNDANT SLOTS AND WATTS—ALL YOU'LL EVER NEED TO CREATE A RANGE OF POWERFUL SERVER APPLICATIONS.

XPRESS DESKSIDE/MX PLATFORMS ARE YOUR FAST TRACK TO THE FUTURE.

SLOTS AND WATTS FOR EXPANSION

Xpress deskside/MX platforms have been designed for OEMs and custom solution providers to integrate the most demanding server applications with maximum investment protection. Integrated I/O (2 channel fast SCSI-2, DMA IDE, two serial ports, one parallel, floppy, mouse, keyboard and high resolution graphics) resides on the baseboard providing increased reliability while enabling the use of all eight EISA bus master slots.



SNAP-IN PERIPHERAL BAYS MAKE IT EASY TO INTEGRATE XPRESS SERVERS

The scaleable Xpress baseboard is housed in an easy-to-integrate, easy-to-service tower which—like the system board—will remain state of the art for years to come. The chassis has been designed to allow ample mass storage capacity and easy access. Two screws release the side of the chassis. A SCSI distribution board connects with a SCSI 2 controller or the optional Xpress RAID adapter and easily distributes power and signal lines to each of the 11 drive bays. No tangle of cables or worry about drive termination. Special snap-in rails let you open the chassis, slide in a drive and connect it in just seconds!

Eight 3.5″ internal expansion bays allow over 12 Gigabytes of storage. Three additional 5.25″ external bays are available for floppies, tapes, CDs, or other devices. And, the power supply is a hefty 384 watts, more power than most, ensuring that future add-in power requirements are easily met. Other Xpress/MX server features:

- Three levels of built-in security.
- Automatic remapping of bad memory.
- Flash memory for easy BIOS upgrades and error logging.
- A user-programmable LCD display.
- Complete diagnostic package.
- Uplift to two year factory warranty.

Additionally, several server accessories are available to complement our Xpress platforms:

- RAID controller for fault tolerant I/O.
- 256 MB Parity memory expansion card
- 384 MB Error checking and correcting memory card (Q4-93)

CHOOSE YOUR OPERATING SYSTEM

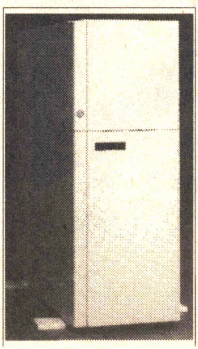
Xpress is an investment in the future. The 32-bit operating systems including Windows NT, Solaris, SCO MPX, Netware SFTIII, Univel, and others will benefit from the same peak performance you can see in today's environments, like Netware, Banyan, SCO UNIX, OS/2 2.0, or Microsoft Windows.

QUALITY YOU CAN SEE

The Xpress Deskside chassis is specially engineered to accommodate future processor modules—with a large power supply, extra cooling, and emissions shielding which will be needed to run higher frequency CPUs.

Intel design and manufacturing is your assurance that Xpress Deskside platforms are reliable—a critical need for server applications. Intel's engineers' attention to detail and our modern factories ensure outstanding quality:

- Protection devices on serial I/O ports avoid voltage spikes which could damage the system.
- Metal tab SIMM sockets ensure solid connections.
- EMI and Thermal guard bands ensure ongoing regulatory compliance

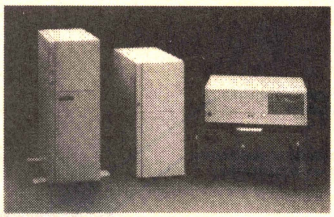


THE XPRESS DESKSIDE PLATFORM WAS DESIGNED FROM THE GROUND UP TO BE A POWER SERVER.

COMPATIBILITY, SERVICE AND SUPPORT FOR OEMS

Our support services are geared to the unique needs of OEMs and custom solution providers. IntelTechDirect™ helps you get to market faster and keep all of your Intel platform products running smoothly. The service includes monthly product updates, FaxBACK™ service, factory hotline, and an electronic bulletin board.

FAMILY OF CHASSIS



To provide complete flexibility, Xpress platforms are available in a range of different form factors—each utilizing the same scalable baseboard technology. In addition to the Deskside/MX tower, Xpress is available as a desktop chassis, a mainstream deskside server, a rack mountable chassis, or as a board level solution.

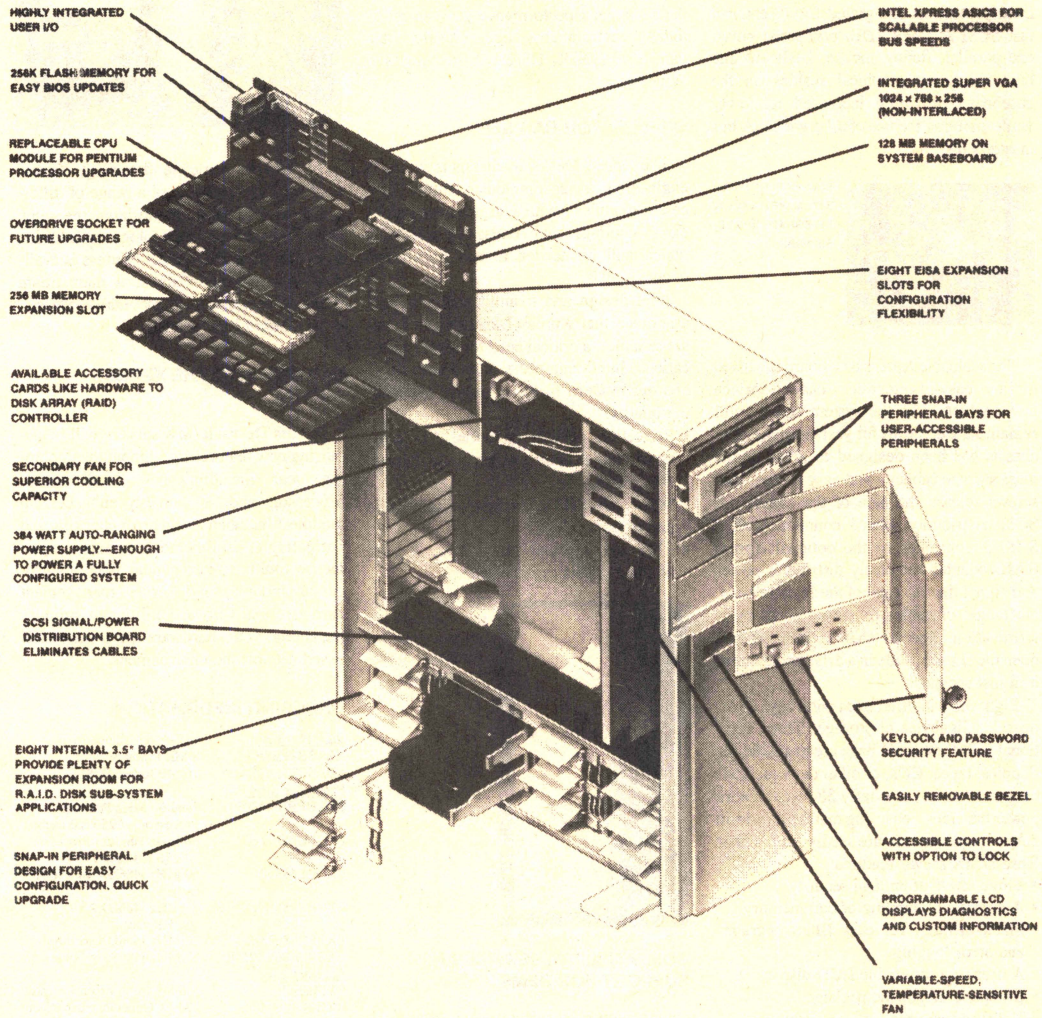
GLIMPSE INTO THE XPRESS FUTURE

Xpress Deskside/MX servers will be acquiring new features throughout the upcoming year including new Intel microprocessors and functionality enhancement modules. Included are an error checking and correcting (ECC) memory module, a symmetric multiprocessing dual pentium processor module, and server management hardware and software utilities to simplify managing your hardware in a standards based networking environment.

ORDERING INFORMATION:

Base Platform	CPU mod. shipped sep.
XBASE8TE8F-C	8 MB tower, SCSI-2
XBASE8TE16F-C	16 MB tower, SCSI-2
Processor Module	
BXCUPBIPENT60	Dual 60 MHz Pentium Processor w/ 256 KB cache
BXCPUPENT60	60 MHz Pentium Processor w/ 256 KB cache
BXCPU48650	50 MHz i486DX w/ 256 KB cache
BXCPU486DX266-B	66 MHz i486DX2 w/ 128 KB cache
BXCPU48633-B	33 MHz i486DX module
BXCPU486SX25-B	25 MHz i486SX module
Accessories	
BXMEM0	256 MB mem. expansion card
BXECCMEM0	384 ECC memory expansion card
XEXTSCSICBL	Cable routes SCSI signal to external device
XRAID4	2 channel RAID controller, 4 MB cache
Baseboard	
BXBASE8E0-C	0 MB baseboard with SCSI-2

XPRESS DESKSIDE/MX PLATFORM



SPECIFICATIONS

SYSTEM BASEBOARD

System Baseboard Memory:	
Architecture:	64 bit, interleaved
Peak Transfer Bandwidth:	133 MB/s
Standard RAM	8 MB (4 MB SIMMs, 2 sockets open)
Maximum RAM	128 MB (32 MB SIMMs, 8 Mbx36)
Access Time	80ns (or faster)
Error Detection	Byte Parity
Total memory:	
Expansion module	256 MB (32 MB SIMMs)
System baseboard	128 MB (32 MB SIMMs)
Total	384 MB
I/O Expansion Slots:	
Eight 32-bit Fast-EISA:	Eight bus master slots
Integrated I/O:	
2 serial ports	Async, RS-232-C, 9-pin
1 parallel port	Centronics and PS/2-compatible, 25-pin
1 mouse port	PS/2-compatible
1 keyboard port	PS/2-compatible
Integrated I/O Controllers:	
Fixed Disk Interface (future models)	Adaptec Fast SCSI 2 (7770-2 channels)
Fixed Disk Interface	IDE (2 drives)
DMA	
Floppy Drive Interface	82077/SL (2 drives)
System BIOS Interface	Intel 28F001BT 256KB Flash
Integrated Graphics:	
Controller	Western Digital WD90C31
Video Memory	512 KB (60ns DRAM) (sockets for easy upgrade to 1 MB DRAM
Maximum Resolutions	800×600, 32,000 colors, "True Color" 1024×768, 256 colors, non-interlaced 1280×1024, 16 colors, non-interlaced
Compatible with:	MDA, CGA, HGC, EGA, VGA, EVGA
Output ports:	15-pin, analog Internal 8514/A Video feature connector (VESA-compatible)
LCD Display:	40 character programmable
Board Dimensions:	
Baseboard	Width 12.45″ (31.62 cm) Depth 12.70″ (32.26 cm)
CPU Module (Some modules vary)	Depth 12.08″ (30.7 cm) Height 4.48″ (11.4 cm)

CPU SPECIFICATIONS, MODULES (CHOICE OF):

Dual Pentium/60 Module:	2x Pentium microprocessors at 60 MHz
Cache:	2x 82495 DX; write-back mode with 2x 256 KB SRAM (82490 DX)
Pentium Processor Modules:	Pentium microprocessors at 60 MHz 82495 DX; write-back mode with 256 KB SRAM (82490 DX)
i486 DX/50 Module:	i486DX microprocessor at 50 MHz
Cache:	82495 DX; write-back mode with 256 KB SRAM (82490 DX)
i486 DX2/66 Module:	i486 DX2 microprocessor at 66 MHz
Cache:	8 KB internal+128 KB L2 write-through
i486 DX/33 Module:	i486 DX microprocessor at 33 MHz
Cache:	8 KB internal+optional 128 KB L2
i486 SX/25 Module:	i486 SX microprocessor at 25 MHz
Cache:	8 KB internal+optional 128 KB L2

MEMORY EXPANSION MODULE

Model BXMEM0:	
Standard RAM	0 MB
Maximum RAM	256 MB (32 MB SIMMs, 8 Mb×36)
Access Time	80 ns
Error detection	Byte Parity

SYSTEM SPECIFICATIONS:

Floppy Disk:	
Format	3.5″, half-height
Capacity	1.44 MB
Drive Bays:	
External	Three 5.25″ width, half-height
Internal	Eight 3.5″ width, full-height
System Dimensions:	
Height	24.40″ (61.98 cm)
Width	8.5″ (21.54 cm)
Depth	23.15″ (58.80 cm)
Weight	54 lbs. (24.5 kg)

ELECTRICAL

AC Voltage/Frequency	Autoranging switching power supply, 100–120/200–240, 60/50 Hz
DC Power	384 W
+5V	45 A maximum continuous
+12V	13 A maximum continuous
−12V	1.0 A maximum continuous
−5V	0.5 A maximum continuous

REGULATIONS

Safety
UL 1950, 1st edition
CAS C22.2 No. 950-89
TUV Bauart to EN 60 950/IEC950/ZHI 618
NEMKO EN60 950

EMI
FCC Class B CFR #47 Parts 2 and 15, Certified Class B
VDE 087/6.78 Level B (Vfg 1046)
CISPR 22/85 Class B, VCCI Class 2
VCCI Class 2 ITE
DOC CRC c.1374 Class B
EN55 101 Series

ENVIRONMENT

Temperature:	
Operating	+10°C to +35°C ambient
Non-operating	−40°C to +65°C ambient
Humidity (Non-condensing wet-bulb max 29°C)	
Operating	20% to 80% @ 35°C
Non-operating	20% to 92% @ 65°C
Altitude:	
Operating	10,000 ft. (3048 m)
Non-operating	50,000 ft (15,240 m)
Vibration:	Random variation 0 Hz–500 Hz
Noise:	<45 db (peripherals idle, at 1 m) @ 25°C
Static discharge:	Tested to 25 kV; no component damage

ACCESSORIES:

Systems include X-Series Deskside/MX Product Guide, EISA Configuration, Utility (ECU) software, video software drivers, SCSI power & distribution board, drive rails for additional I/O devices, power cable, floppy and IDE hard drive cables, system diagnostics and a warranty uplift option for two years return to factory coverage.

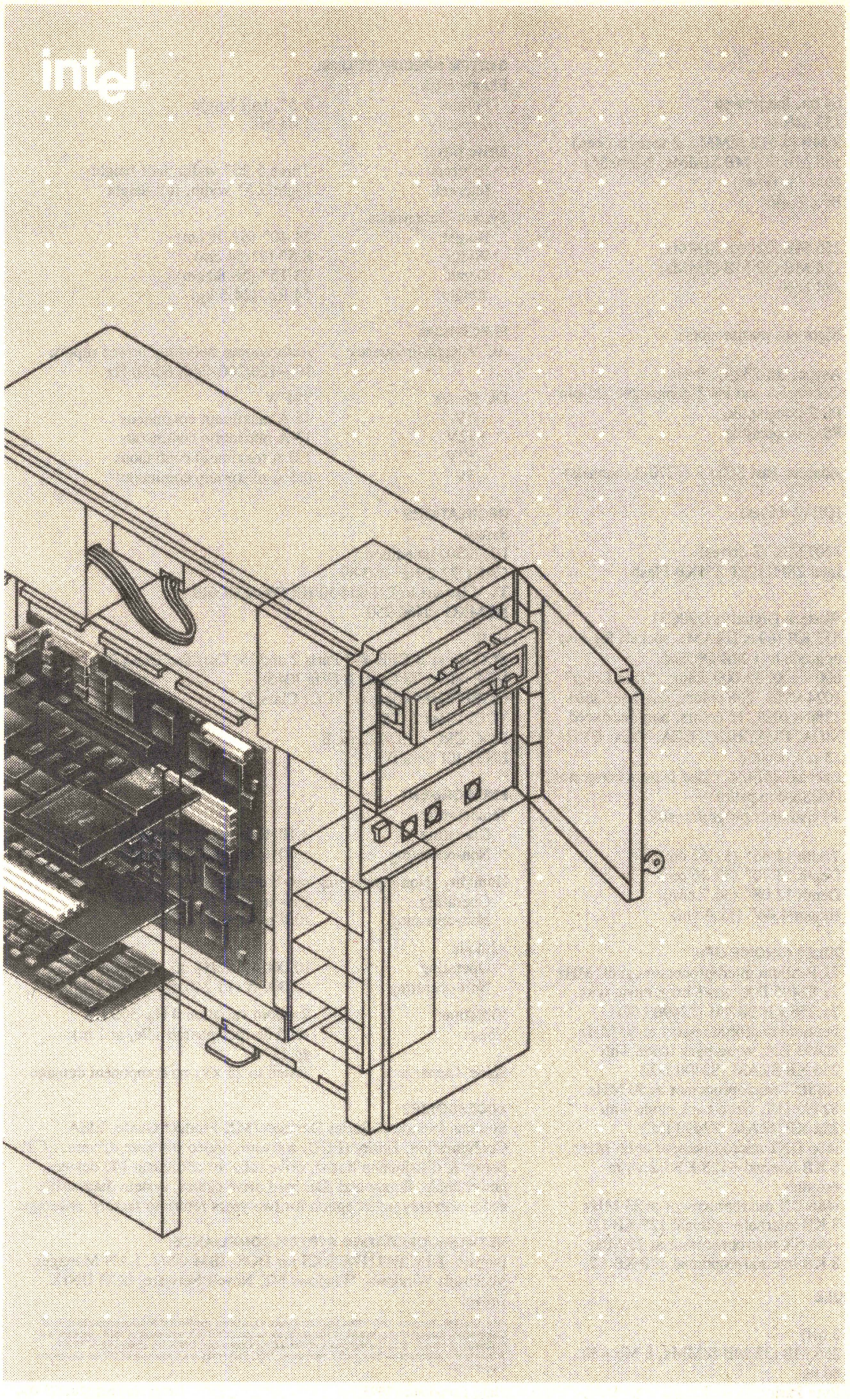
NETWORK/OPERATING SYSTEM COMPLIANCE:

Banyan, DEC PATHWORKS for DOS, IBM OS/2, LAN Manager, Microsoft Windows, Windows NT, Novell Netware, SCO UNIX, Univel

i486, Intel486, Pentium, DVI, i386, IntelTechDirect, OverDrive and FaxBACK are trademarks of Intel Corporation. Banyan is a trademark of Banyan Systems Inc., OS/2 is a trademark of IBM, Netware is a trademark of Novell, SCO is a trademark of Santa Cruz Operations, UNIX is a trademark of AT&T, Windows is a trademark of Microsoft Corporation, DEC Pathworks is a trademark of Digital Equipment Corporation.

©Intel Corporation 1993

Order Number: 281379-004



INTEL XPRESS/LX DESKSIDE PLATFORMS

i486™ AND PENTIUM™ MICROPROCESSORS

October 1993
Order Number: 281558-001

AN AFFORDABLE SERVER WITH THE RIGHT FEATURES

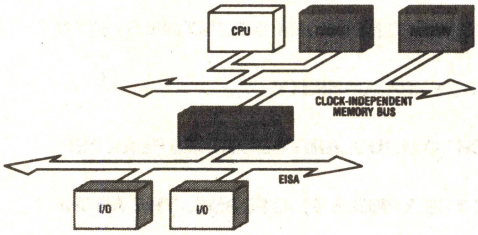
XPRESS/LX IS A LEADING EDGE MODULAR SERVER SOLUTION TARGETED FOR THE MAINSTREAM SERVER MARKET. THE XPRESS/LX DESIGN IS IDEAL FOR LOCAL AREA NETWORKS OR POWER USERS REQUIRING A VERTICAL FOOTPRINT, INCLUDING WORKGROUP, FILE, COMMUNCATION AND APPLICATION SERVICES. IT REPRESENTS A NEW ENTRY POINT FOR SERVERS WITH A ROBUST FEATURE SET, COMPLETE NETWORK OPERATING SYSTEM COMPATIBILITY AND FUTURE EXPANDABILITY.

AN INGENIOUS APPROACH TO MODULARITY ENSURES PEAK PERFORMANCE FROM ALL OF THE XPRESS PLATFORMS. THE ARCHITECTURE ALLOWS YOU TO SWAP IN MORE POWERFUL MICROPROCESSORS AS THEY BECOME AVAILABLE. TODAY'S XPRESS/LX DESKSIDE CHASSIS WILL TAKE YOU FROM 20 TO OVER 100 MIPS IN A SINGLE BOX! YOUR CUSTOMER SIMPLY REPLACES THE OLD CPU MODULE AND SPEEDS TO MARKET WITHOUT TIME-CONSUMING RECONFIGURATION AND REQUALIFICATION.

XPRESS/LX DESKSIDE PLATFORMS ARE YOUR FAST TRACK TO THE FUTURE.

Xpress Deskside/LX provides robust server performance and upgradeable features to entry-level customers. The easily replaceable CPU modules offer a range of speeds that can exceed 100 MIPs using today's Pentium™ microprocessor. And these servers are designed to provide peak performance for years to come because they're ready for even faster speeds using next generation CPUs and I/O technology.

The latest Intel microprocessors can be used instantly in Xpress platforms. The unique modular architecture isolates the fastest evolving technology – the CPU and cache – on a replaceable card. This benefits OEMs and Custom Solution Providers because it simplifies product design, reduces sales development requirements, and provides customers with a way to economically upgrade their systems.

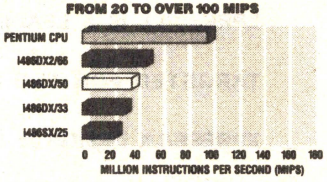


DESIGNED FOR THE FUTURE

Xpress platforms are not only modular, they're scalable, accommodating both today's processors and tomorrow's as well. The 64-bit interleaved memory architecture provides exceptionally high transfer speeds (133 Mbytes per second peak transfer). Since the CPU-memory access is clock-independent, Xpress is able to support tomorrow's CPUs of up to 100 MHz *at full performance*, and as a plus, the memory architecture supports industry standard fast page 1 MB through 32 MB SIMMs for maximum flexibility and affordability.

With a range of single processor modules, and next generation microprocessor modules on the horizon, Xpress/LX provides server class performance of over 100 MIPS! The result is a world-class server platform which allows our customers to demonstrate award winning server performance today, and which can provide peak performance for years to come.

Xpress/LX servers offer a range of performance points with plenty of room for expansion. Module options provide from 20 to over 100 MIPS today – with more to come from next generation's processors.

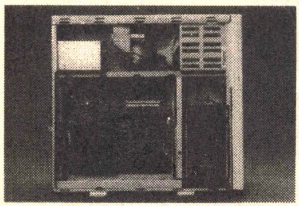


There are three types of Xpress CPU modules to choose from:

- **Transaction Processing and Server.** Tuned to provide the highest possible server performance, the DX/50 and Pentium microprocessor modules include a 256 KB write-back cache which provides up to 36% higher performance than traditional write-through or direct-map cache solutions. This reduces CPU utilization of the memory bus and enables concurrent cache and I/O operations, resulting in enhanced performance in multi-user I/O intensive systems such as network or dedicated application servers. For even greater performance, next generation Intel processors will be supported by today's Xpress platforms.
- **Compute Intensive.** Designed with the latest Intel silicon technology, Intel486™ DX2 modules use internal clock doubling techniques to provide the highest performance for compute-intensive applications. Complete with 128 KB cache and a socket for Intel's Future OverDrive™ processor, these modules are ideal for advanced computer simulations or distributed processing applications.
- **Entry Level Modules.** Providing optional cache support and an Overdrive processor socket, these modules are designed to give an economical entry to the Xpress platforms and are ideal for file servers or customers upgrading their existing i386™ architecture designs.

Xpress/LX deskside platforms have been designed for OEMs and custom solution providers to integrate demanding server applications with maximum investment protection. The scaleable Xpress system board will remain state of the art for years to come, along with the easy-to-integrate, easy-to-service tower enclosure.

The chassis has been designed to allow ample mass storage capacity and easy access. Two screws release the side of the chassis. Five 3.5" internal expansion bays allow for ample storage. Special snap-in rails let you open the chassis, slide in a drive and connect it in just seconds! Three additional 5.25" external bays are available for floppies, tapes, CDs, or other devices. Integrated I/O (DMA IDE, two serial ports, one parallel, floppy, mouse, keyboard and high resolution graphics) resides on the baseboard providing increased reliability while enabling the use of all eight EISA bus master slots.



XPRESS/LX IS PACKAGED IN A SPACE-SAVING TOWER WITH DESIGNED IN SCALABILITY TO NEXT GENERATION INTEL PROCESSORS.

Other Xpress/LX server features:

- Three levels of built-in security – Master key option available.
- Flash BIOS for easy upgrades.
- Robust design for embedded applications
- Optional diagnostic package.

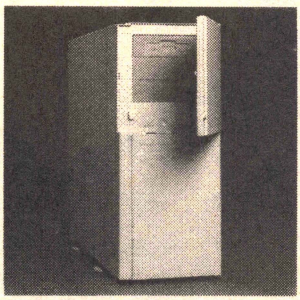
CHOOSE YOUR OPERATING SYSTEM

Xpress is an investment in the future. The 32-bit operating systems including Windows NT, Solaris, SCO MPX, Netware SFTIII, Univel, and others will benefit from the same peak performance you can see in today's environments, like Netware, Banyan, SCO UNIX, OS/2 2.0, or Microsoft Windows.

The Xpress/LX Deskside chassis is specially engineered to accommodate future processor modules with extra cooling and emissions shielding which will be needed to run higher frequency CPUs.

Intel design and manufacturing is your assurance that Xpress Deskside platforms are reliable – a critical need for server applications. Intel's engineers' attention to detail and our modern factories ensure outstanding quality:

- Self-resetting fuse protects keyboard from voltage spikes
- Rounded metal edges for smooth integration of peripherals
- EMI and Thermal guard bands ensure ongoing regulatory compliance

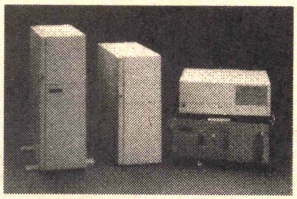


COMPATIBILITY, SERVICE AND SUPPORT FOR OEMS

Our support services are geared to the unique needs of OEMs and custom solution providers. IntelTechDirect™ helps you get to market faster and keep all of your Intel platform products running smoothly. The service includes monthly product updates, FaxBack service, factory hotline, and an electronic bulletin board.

FAMILY OF CHASSIS

To provide complete flexibility, Xpress platforms are available in a range of different form factors – each utilizing the same scalable baseboard technology. In addition to the Deskside/LX server, Xpress is available as a desktop chassis, a premium server tower, a rack mountable chassis, or as a board level solution.



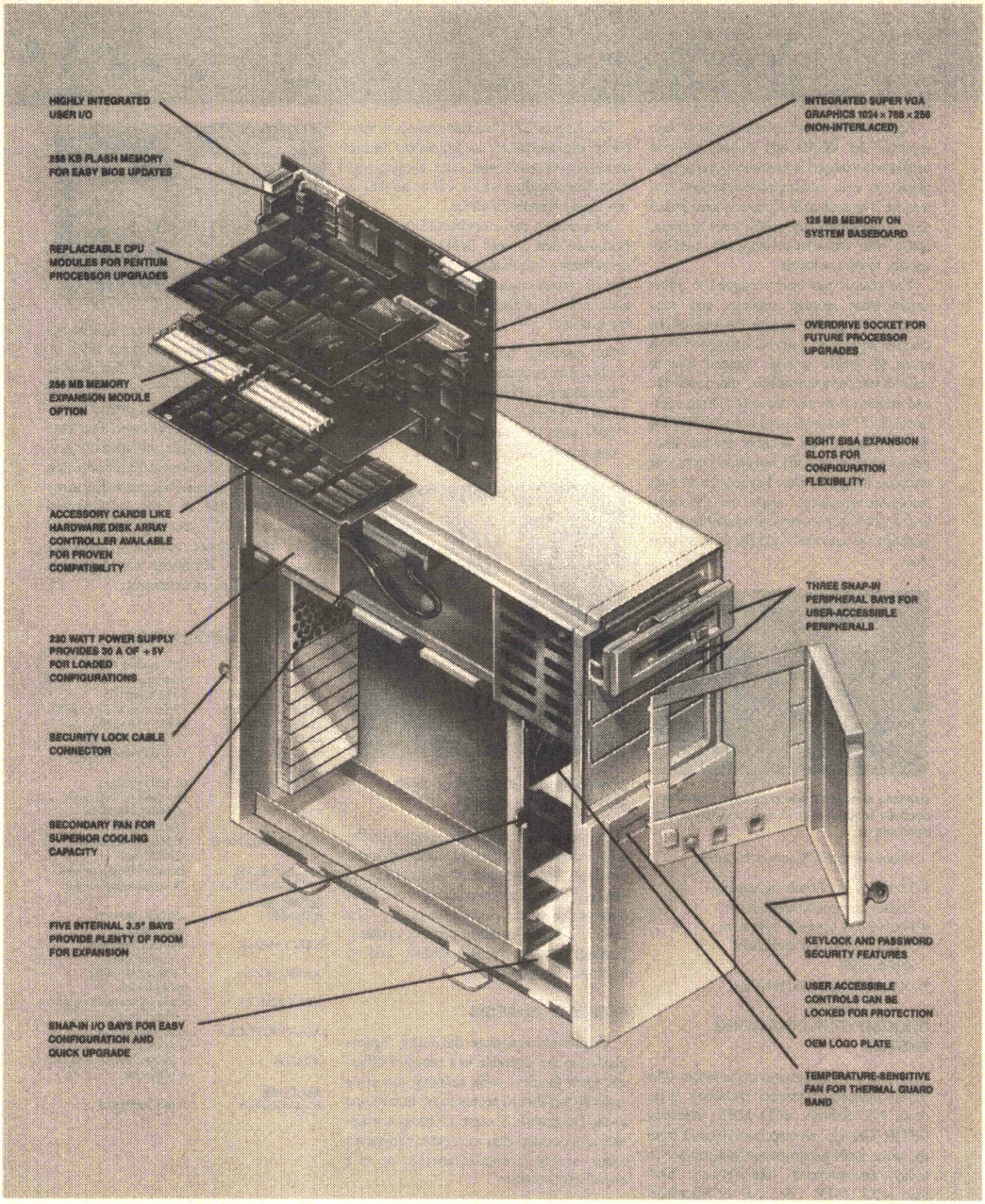
Xpress Deskside/LX servers will be acquiring new features throughout 1993. A second model with Adaptec's fast SCSI-2 controller integrated on the baseboard will be introduced in late summer providing enhanced performance at a lower total cost than add-in adapter cards. Additionally, several accessories are planned for introduction including a two channel hardware disk array (RAID) controller, an error checking and correcting (ECC) memory module and server management utilities to simplify managing your hardware in a standards based networking environment.

ORDERING INFORMATION:

Base Platform	CPU module shipped separately
XLXBASE8TE8F	8 MB tower (US)
XLXBASE8TE8FS	8 MB tower (US) w/ SCSI
XLXBASE8TE16F	16 MB tower (US)
XLX8E8F220V	8 MB tower (Europe)
XLX8E16F220V	16 MB tower (Europe)
Processor Module	
BXCPUPENT60	60 MHz Pentium Processor w/ 256 cache
BXCPU48650	50 MHz i486DX w/ 256 KB cache
BXCPU486DX266-B	66 MHz i486DX2 w/ 128 KB cache
BXCPU48633-B	33 MHz i486DX module
BXCPU486SX25-B	25 MHz i486SX module
Accessories	
BXMEM0	256 MB memory expansion card
BXECCMEM0	384 ECC memory expansion card
XLXSCSICBL	Cable kit for SCSI configuration
XEXTSCSICBL	Cable routes SCSI signal to external device
XDIAGNOSTICS	Complete system diagnostics
XRAID4	2 channel RAID controller, 4 MB cache
Baseboard	
BXBASE8E0-B	0 MB baseboard

16

XPRESS DESKSIDE/LX PLATFORM



SYSTEM BASEBOARD

System Baseboard Memory:	
Architecture:	64 bit, interleaved
Peak Transfer Bandwidth:	133 MB/s
Standard RAM	8 MB (4 MB SIMMs, 2 sockets open)
Maximum RAM	128 MB (32 MB SIMMs, 8 Mb×36)
Access Time	80ns (or faster)
Error Detection	Byte Parity
Total memory	
Expansion module	256 MB (32 MB SIMMs)
System baseboard	128 MB (32 MB SIMMs)
Total	384 MB
I/O Expansion Slots:	
Eight 32-bit Fast-EISA:	Eight bus master slots
Integrated I/O:	
2 serial ports	Async, RS-232-C, 9-pin
1 parallel port	Centronics and PS/2-compatible, 25-pin
1 mouse port	PS/2-compatible
1 keyboard port	PS/2-compatible
Integrated I/O Controllers	
Fixed Disk Interface (future models)	Adaptec Fast SCSI 2 (7770-2 channels)
Fixed Disk Interface	IDE (2 drives)
Floppy Drive Interface	82077/SL (2 drives)
System BIOS Interface	Intel 28F001BT 256KB Flash
Integrated Graphics	
Controller	Western Digital WD90C31
Video Memory	512 KB (60ns DRAM) (sockets for easy upgrade to 1 MB DRAM
Maximum Resolutions	800×600, 32,000 colors, "True Color" 1024×768, 256 colors, non-interlaced 1280×1024, 16 colors, non-interlaced
Compatible with:	MDA, CGA, HGC, EGA, VGA, EVGA
Output ports:	15-pin, analog Internal 8514/A Video feature connector (VESA-compatible)
Board Dimensions:	
Baseboard	Width 12.45″ (31.62 cm) Depth 12.70″ (32.26 cm)
CPU Module (Some modules vary)	Depth 12.08″ (30.7 cm) Height 4.48″ (11.4 cm)

CPU SPECIFICATIONS, MODULES (CHOICE OF):

Pentium Processor Modules:	Pentium microprocessors at 60 Mhz 82495 DX; write-back mode with 256 KB SRAM (82490 DX)
i486 DX/50 Module:	i486DX microprocessor at 50 MHz
Cache:	82495 DX; write-back mode with 256 KB SRAM (82490 DX)
i486 DX2/66 Module:	i486 DX2 microprocessor at 66 MHz
Cache:	8 KB internal+128 KB L2 write-through
i486 DX/33 Module:	i486 DX microprocessor at 33 MHz
Cache:	8 KB internal+optional 128 KB L2
i486 SX/25 Module:	i486 SX microprocessor at 25 MHz
Cache:	8 KB internal+optional 128 KB L2

MEMORY EXPANSION MODULE

Model BXMEM0:	
Standard RAM	0 MB
Maximum RAM	256 MB (32 MB SIMMs, 8 Mb×36)
Access Time	80 ns
Error detection	Byte Parity

SYSTEM SPECIFICATIONS:

Floppy Disk	
Format	3.5″, half-height
Capacity	1.44 MB
Drive Bays	
External	Three 5.25″ width, half-height
Internal	Five 3.5″ width, full-height
System Dimensions	
Height	20.4″ (51.81 cm)
Width	8.5″ (21.6 cm)
Depth	22.5″ (57.12 cm)
Weight	44.51 lbs. (20.19 kg)

ELECTRICAL

AC Voltage/Frequency	100–120/200–240, 60/50 Hz User selectable
DC Power	230 W
+5V	30 A maximum continuous
+12V	8 A maximum continuous
−12V	0.5 A maximum continuous
−5V	0.5 A maximum continuous

REGULATIONS

Safety:
UL 1950, 1st edition
CAS C22.2 No. 950-89
TUV Bauart to EN 60 950/IEC950/ZHI 618
NEMKO EN60 950

EMI:
FCC Class B CFR #47 Parts 2 and 15, Certified Class B
VDE 087/6.78 Level B (Vfg 1046)
CISPR 22/85 Class B, VCCI Class 2
VCCI Class 2 ITE
DOC CRC c.1374 Class B
EN55 101 Series

ENVIRONMENT

Temperature:	
Operating	+10°C to +35°C ambient
Non-operating	−40°C to +65°C ambient
Humidity (Non-condensing wet-bulb max 29°C)	
Operating	20% to 80% @ 35°C
Non-operating	20% to 92% @ 65°C
Altitude:	
Operating	10,000 ft. (3048 m)
Non-operating	50,000 ft. (15,240 m)
Vibration:	Random variation 0 Hz–500 Hz
Noise:	<42 db (peripherals idle, at 1 m) @ 25°C
Static discharge:	Tested to 25 kV; no component damage

ACCESSORIES:

Systems include: X-Series Deskside/LX Product Guide, EISA Configuration Utility (ECU) software, video software drivers, drive rails for additional I/O devices, power cable, floppy and IDE hard drive cable, a one year warranty.

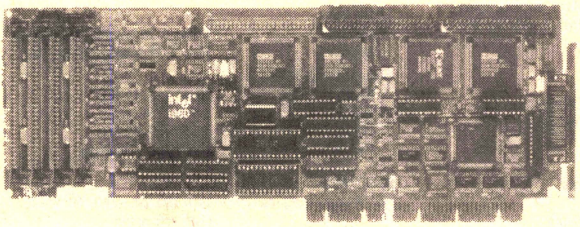
NETWORK/OPERATING SYSTEM COMPLIANCE:

Banyan, DEC PATHWORKS for DOS, IBM OS/2, LAN Manager, Microsoft Windows, Windows NT, Novell Netware, SCO UNIX, Univel

i486, Intel486, Pentium, DVI, i386, IntelTechDirect and OverDrive are trademarks of Intel Corporation. Banyan is a trademark of Banyan Systems Inc., OS/2 is a trademark of IBM, Netware is a trademark of Novell, SCO is a trademark of Santa Cruz Operations, UNIX is a trademark of AT&T, Windows is a trademark of Microsoft Corporation, DEC Pathworks is a trademark of Digital Equipment Corporation.

©Intel Corporation 1993

Order Number: 281558-001



XPRESS DISK ARRAY CONTROLLER HIGHER I/O PERFORMANCE & DATA LOSS PROTECTION

High performance compute platforms, like the clock doubling i486™ DX/2 and Pentium™ microprocessors, require faster disk subsystems. To meet this increased performance need, and to satisfy the need for protection against data loss, Intel is now offering a two channel, SCSI RAID (Redundant Array of Inexpensive Disks) solution to complement the Xpress platform line of scalable servers. Intel's EISA disk array controller, based on the Intel i960™ microprocessor, provides a high performance hardware RAID solution supporting data striping for performance, mirroring for data redundancy, and distributed striping with parity for the highest data availability at minimal cost.

Designed for network server, application server, and multi-user UNIX* environments, Intel's disk array controller can improve data throughput, balance I/O performance, and ensure availability of critical data. Best of all, the disk array controller has been tested and certified for SCO Unix and Novell Netware* environments to ensure the quality and reliability you expect from Intel.

FAMILY OF PLATFORMS

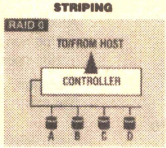
The Xpress disk array controller is one of many accessories designed specifically for Xpress platforms. The base platforms are available in a range of different form factors—each utilizing the same scalable baseboard technology which supports Intel486™ to Pentium processors. Chassis options include an expandable desktop, a mainstream deskside server (Xpress/LX), a rack mountable chassis, and a premium server tower (Xpress/MX). Xpress servers are designed with snap-in drive bays to reduce integration time. The Deskside/MX includes a SCSI signal and power distribution board designed to eliminate the need for long cables and increase signal reliability when used with the Xpress disk array controller.

COMPATIBILITY, SERVICE AND SUPPORT FOR OEMS

Our support services are geared to the unique needs of OEMs and custom solution providers. IntelTechDirect™ helps you get to market faster and keep all of your Intel platform products running smoothly. The service includes monthly product updates, FaxBACK™ service, factory hotline, and an electronic bulletin board.

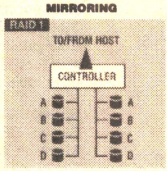
RAID LEVEL 0,1,5 HARDWARE SUPPORT

STRIPING



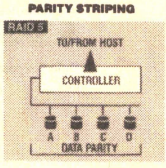
BLOCK STRIPING SEGMENTS DATA BETWEEN SEVERAL DISKS, PROVIDING HIGHER PERFORMANCE THAN WITH INDIVIDUAL DRIVES

MIRRORING



DRIVES CAN BE PAIRED AND MIRRORED PROVIDING 100% DATA DUPLICATION FOR REDUNDANCY

PARITY STRIPING



DATA IS STRIPED AND DISTRIBUTED WITH PARITY ACROSS SEVERAL DISKS TO PROVIDE HIGH DATA AVAILABILITY WITH FEWER DISKS (LOWER COST) THAN MIRRORING

November 1993
Order Number: 281559-002

Feature	Function	Benefit
Two I/O channels	Supports up to 14 SCSI drives	Highest possible performance with industry standard compatibility
Fast SCSI-2	10 MB/s data transfer	Leading edge performance
Intel i960 processor	32 bit RISC processor	Highest possible performance
Support for standard I/O devices	Support for high performance disk drives, CD-ROM, tape, and optical drives	Industry compatibility
Support for: RAID Level 0 RAID Level 1 RAID Level 5 Other combinations	 Striping Mirroring Striping w/ distributed parity 0/1 and spanning	 Performance Redundancy Performance/Availability Flexibility
Flash firmware and BIOS	Programmable BIOS	Easy to support and upgrade
Up to 64 MB write back cache	4,16, and 64 MB read/write cache with standard SIMMs	Highest performance at minimum cost
External SCSI-2 connector	External device support	Flexibility
Simultaneous data transfer	Data transfer between I/O bus/controller and controller/disk at same time	Highest performance
Automatic drive failure detection	Constant monitoring of disk array	Instant notification of system drive problems
Hot spare option	Replace failing drive instantly	No unscheduled down time
Automatic drive rebuild	Replaced drive / spare drive will be rebuilt in the background	Rebuild time is optimized based on requests to the I/O subsystem
Complete utilities	Drive card is easy to configure and install	Minimum time investment

SPECIFICATIONS

BOARD

CPU	i960CA 32-bit microprocessor
Cache option	4, 16 or 64 MB
Standard	4 MB
Cache support	Write-through Write-back Cache flush
SCSI support	SCSI-2 fast, 8-bit wide
SCSI channels	2 (7 devices per channel)
SCSI controller	NCR 53C720 PT
EISA bus master	Supports EISA burst mode transfers of up to 33 MB/s
Onboard BIOS	Flash EEPROM (16K×8)
Firmware	Flash EEPROM (16K×8)
RAID Levels	0 - Block striping, no redundancy 1 - Mirroring 5 - Block striping with rotated XOR redundancy Striping & mirroring
Multi-threaded	64 simultaneous commands
Scatter/gather	
Fault tolerance	Automatic drive failure detection Hot spare On-line rebuilding of replacement drive
Power sequencing	

SOFTWARE

OS Support	SCO Unix System V 4.0, 4.2 Novell Netware 3.11, 4.0 Windows NT (version 3.1) IBM OS 2.0 (2.1) Banyan (5.52 (5)

ENVIRONMENT (System Level)

Ambient Temperature:

Operating	+10°C to +35°C ambient
Non-Operating	−40°C to +70°C ambient

Relative Humidity:

Operating	85% @ 35°C
Non-Operating	95% @ 35°C

Altitude:

Operating	to 10,000 ft. (3048 m)
Non-Operating	to 50,000 ft. (15,240 m)
Static Discharge	25 kV no component damage

REGULATIONS

Meets or exceeds the following requirements:

EMI/RFI

U.S.	FCC Class B

TOOLS

Configuration	Menu driven DOS utility
OS Utilities	Utilities for all supported operating systems

ORDERING INFORMATION

Order Code XRAID4	2 Channel SCSI disk array controller with 4 MB cache

For more information or the number of your nearest Intel sales office, please call the numbers below.

UNITED STATES
Intel Corporation
2200 Mission College Dr.
Santa Clara, CA 95052-8119
Tel : 800-548-4725

EUROPE
Intel Corporation (U.K.) Ltd
Pipers Way
Swindon
Wiltshire SN3 1RJ
United Kingdom
Tel: +44 793 696000

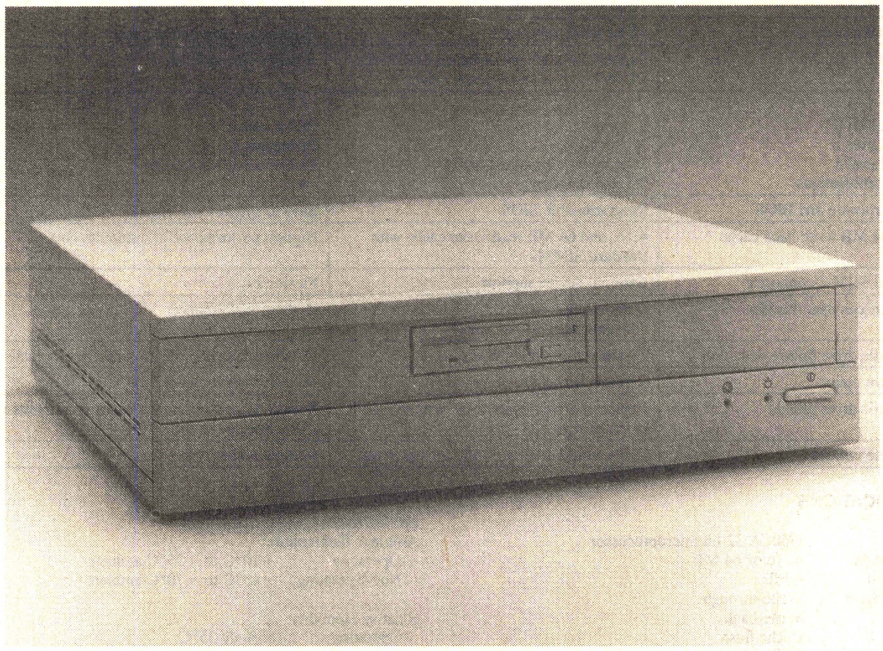
JAPAN
Intel Japan K.K.
5-6 Tokodai, Tsukuba-shi
Ibaraki, 300-26
Tel : +81-29847-8522

i486, Intel486, Pentium, IntelTechDirect, FaxBACK and OverDrive are trademarks of Intel Corporation. Banyan is a trademark of Banyan Systems Inc., OS/2 is a trademark of IBM, Netware is a trademark of Novell, SCO is a trademark of Santa Cruz Operations, UNIX is a trademark of AT&T, Windows is a trademark of Microsoft Corporation.

© INTEL CORPORATION 1993

ORDER # 281559-002

i486™ CHAMPION LOW PROFILE PLATFORM



281580-1

i486™ CHAMPION LOW PROFILE PLATFORM—WITH ACCELERATED LOCAL BUS

The i486™ Champion Low Profile is an entry-level microcomputer platform manufactured by Intel. It offers i486 performance at an affordable price, with built-in accelerated Super VGA Local Bus graphics. The Champion Low Profile protects your hardware investment by supporting Intel OverDrive™ processors, making it easy to upgrade the CPU performance.

FEATURES

- High performance Intel486™ Microprocessors
- OverDrive processor support
- 4 MB on-board memory (expandable to 32 MB)
- Accelerated Super VGA Local Bus graphics integrated on motherboard
- 512 KB video RAM (expandable to 1 MB)
- Integrated IDE hard disk and floppy controller
- Two serial, one parallel, PS/2 keyboard and mouse ports
- Three 16-bit ISA expansion slots
- Three peripheral bays (2 external, 1 internal)
- FCC Class B certification
- Novell* certified

BENEFITS

- Affordable i486 SX/25, DX33 and DX2/66 Microcomputer performance for the desktop
- Provides an easy path for future performance upgrades to protect your hardware investment
- Offers flexibility for a variety of system memory requirements

i486™ and OverDrive™ are trademarks of Intel Corporation.
Novell is a registered trademark of Novell, Inc.

October 1993
Order Number: 281580-002

intel

i486™ CHAMPION LOW PROFILE PLATFORM

BENEFITS

- Excellent support for graphics-intensive applications with both Local Bus and Acceleration
- Optimizes video resolutions to a maximum of 1024 x 768 x 256 colors
- Leaves all slots open for expansion
- Reduces expense of adding peripheral, control and communications devices
- Meets current and future needs for add-in cards
- Flexibility for installing storage and other devices
- Meets rigid requirements for office or home use
- Assured compatibility with Novell networks

SPECIFICATIONS

BOARD

CPU	i486 SX Microprocessor; 25 MHz
	i486 DX Microprocessor; 33 MHz and i486 DX266 Microprocessor, 66 MHz
Performance Upgrade	Intel OverDrive processors; 50 MHz/60 MHz
Graphics Subsystem	Local Bus SVGA Cirrus 5426 integrated
On-board Memory	2 72 pin SIMM sockets
Standard RAM	4 MB (1 MB x 32 SIMM)
Maximum RAM	32 MB (16 MB x 32)
Upgrade RAM	×32 SIMM's
Access Time	80 ns or faster
I/O Expansion Slots	3 16-bit ISA slots (full length)
BIOS Interface	AWARD EPROM
Integrated I/O	
2 serial ports	Async, RS-232C, 9-pin
1 parallel port	Centronics compatible, 25-pin bi-directional
1 keyboard port	PS/2 compatible
Floppy disk controller	Provided
IDE hard disk	Provided
Mouse port	PS/2 compatible

SYSTEM

System Dimensions	
Height	4.10″
Width	16.30″
Depth	15.70″
Drive Bays	
External	1 5.25″ half-height accessible (1.6″)
	1 3.5″ one-third height accessible (1.0″)
Internal	1 3.5″ one-third height (1.0″)
Floppy Disk (standard on every system)	
Footprint	3.5″ one-third height (1.0″)
Capacity	1.44 MB high density
Accessories	
Champion Series Low Profile Product Guide	
Power cable, hard drive cables, and miscellaneous hardware	
Video Drivers Diskette(s)	

ELECTRICAL

AC Voltage/ Frequency	115V/60 Hz, 230V/50 Hz
DC Power Supply	150W
+5V	18A maximum continuous
+12V	4.2A maximum continuous
−5V	0.3A maximum continuous
−12V	0.4A maximum continuous
Available Power	Maximum system power dissipation is 120W

16

intel

SPECIFICATIONS

ENVIRONMENT

Ambient Temperature

Operating	+10°C to +35°C ambient

Relative Humidity

Operating	85% @ 35°C
Non-operating	95% @ 35°C

Altitude

Operating	to 100,000 ft. (3048m)
Non-operating	to 50,000 ft. (15,240m)
Noise	40 db with peripherals idle

REGULATIONS

Meets or exceeds the following requirements

Safety

U.S.	UL 1950
Canada	CSA 950
Europe	TUV to EN60950, IEC 950

EMI/RFI

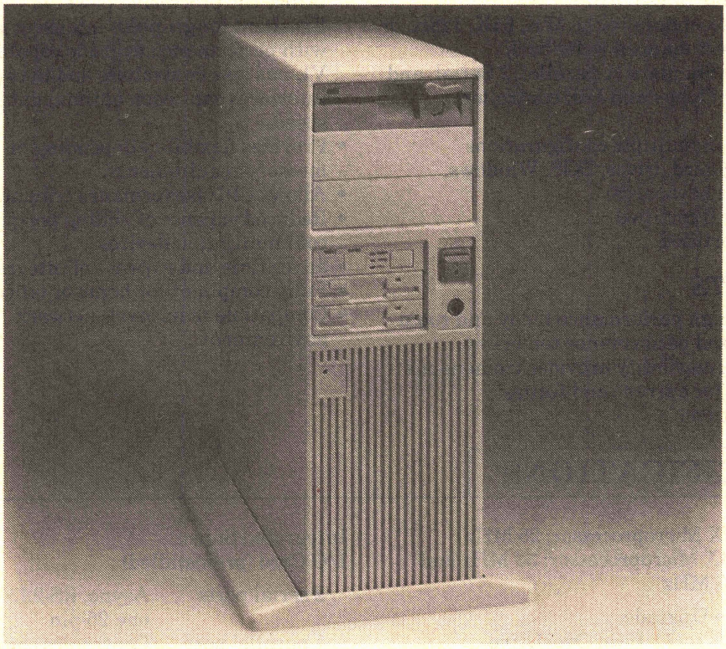
U.S.	FCC CFR 47 Part 15, Class B
Canada	DOC CRC c. 1374 Class B
Europe	VDE 0871 Class B, CISPR-22 Class B

ORDERING INFORMATION

The Champion Low Profile product is available in multiple processor speeds equipped either with 4 MB of memory, a floppy drive and keyboard or as a fully configured system including all of the above items, hard drive DOS, Windows, and Mouse.

Contact your nearest Intel Distributor.

i486™ CHAMPION MID-TOWER PLATFORM



281579-1

i486™ CHAMPION SERIES MID-TOWER PLATFORM WITH VESA LOCAL BUS

The Champion Series Mid-Tower platform features affordable, flexible performance and is manufactured by Intel. The Champion Mid-Tower is available in 3 powerful microprocessor speeds (i486™ SX25, i486 DX33 and i486 DX266). The system CPU is also upgradable using Intel's OverDrive™ processors. The Mid-Tower offers 6 available ISA expansion slots (including 1 VESA Local bus slot), eight peripheral bays in a chassis designed for fast integration and easy servicing. The system is also equipped with a high-performance accelerated SVGA VESA local bus graphic subsystem. The Mid-Tower Champion from Intel is the optimum solution for cost sensitive sever applications requiring the expandability of a tower platform.

16

FEATURES

- SX25/DX33/DX266 i486 Microprocessors and upgrades to Intel's current and future i486 OverDrive Processors

  Tower Case with six available expansion slots (one VL). Eight drive bays, five accessible, (3) 5.25″ and (2) 3.5″

- 2 VESA local bus (VL) slots. One filled with VL VGA graphics accellerator card 1280 x 1024 x 256 colors and 512K video DRAM standard (expandable to 2 MB video DRAM and 1280 x 1024 x 256 colors)

i486™ and OverDrive™ are trademarks of Intel Corporation.
Novell is a registered trademark of Novell, Inc.

October 1993
Order Number: 281579-001

intel

i486™ CHAMPION MID-TOWER PLATFORM

- 8 MB On-board SIMM memory standard—(Expandable to 128 MB)
- Optional secondary cache 32K, 64K, 128K, or 256K (Direct mapped write-back)
- Super I/O Standard (1 parallel, 2 Serial, and keyboard) Floppy and IDE controller standard
- Available in multiple configurations (including hard drives, DOS, Windows, mouse and keyboards)
- FCC Class B certified
- Novell* certified

BENEFITS

- Provides high performance today and room for increased performance tomorrow. Expansion capability provides tremendous flexibility for current and future configurations.
- Local Bus enables VL add-in cards to fully utilize the processing power of the CPU. Excellent for graphics intensive applications with performance upgrade capabilities. VL-bus disk controllers and network interfaces can reach optimal data transfer speeds.
- Provides flexibility depending on system memory requirements.
- Allows CPU performance enhancements.
- Reduced expense of adding peripheral and communication devices.
- Saves time and expense of integration.
- Fully compliant for home or office.
- Compatible in a Novell network environment.

SPECIFICATIONS

CPU	i486 SX Microprocessor; 25 MHz i486 DX Microprocessor; 33 MHz and DX266 MHz
Performance Upgrade	Intel OverDrive processors; 50/66 MHz future Intel OverDrive processor for i486 DX266 (238 pin socket)
Graphic Subsystem	Cirrus 5426 Standard
VESA Add-in Card preinstalled	S3 805 Optional
Cache Option (Direct Mapped write-back)	32 KB/64 KB/128 KB/256 KB upgrade option using 20 ns SRAMs
On-board Memory	8 30-pin SIMM sockets
Standard RAM	8 MB (1 Mb x 9 SIMM)
Maximum RAM	128 MB (16 Mb x 9)
Upgrade RAM	256 Kb x 9, 1 Mb x 9, 16 Mb x 9
Access Time	80 ns or faster
I/O Expansion Slots	8 ISA expansion slots total 2 VESA Local Bus/16-bit ISA slots 4 16-bit ISA slots 2 8-bit ISA slots
BIOS Interface	AWARD EPROM

Integrated Super I/O Card (preinstalled)

2 serial ports	Async, RS-232C, one 9-pin/one 25-pin
1 parallel port	Centronics compatible, 25-pin bi-directional
1 keyboard port	AT compatible
Floppy disk controller	1.44 MB on-board
IDE hard disk connector	On-board
Board Dimensions	8.5″ x 13″, four layers

System Dimensions

Height:	18.5″
Width:	6.75″
Depth:	15.75″

Drive Bays

External	3 5.25″ half-height accessible (1.6″) 2 3.5″ one-third height accessible (1.0″)
Internal	3 3.5″ one-third height accessible (1.0″)

Floppy Disk (standard on every system)

Footprint	3.5″ one-third height (1.0″)
Capacity	1.44 MB high density

Accessories

Champion Series Mid-Tower Product Guide
Power cable, IDE hard drive cables, and miscellaneous H/W
Video Driver Diskette(s)

intel

SPECIFICATIONS

ELECTRICAL

AC Voltage/Frequency	115V/60 Hz, 230V/50 Hz
DC Power Supply	230 W
+5V	30A maximum continuous
+12V	8A maximum continuous
−5V	0.5A maximum continuous
−12V	0.5A maximum continuous
Available Power	For ISA slots (in an integrated system)
+5V	27A
+12V	7A
−12V	0.3A
−5V	0.3A

ENVIRONMENT

Ambient Temperature	
Operating	+10°C to +35°C ambient
Relative Humidity	
Operating	85% @ 35°C
Non-operating	95% @ 35°C
Altitude	
Operating	to 10,000 ft. (3048m)
Non-operating	to 50,000 ft. (15,240m)
Noise	42 dB with peripherals idle

REGULATIONS

Meets or exceed the following requirements

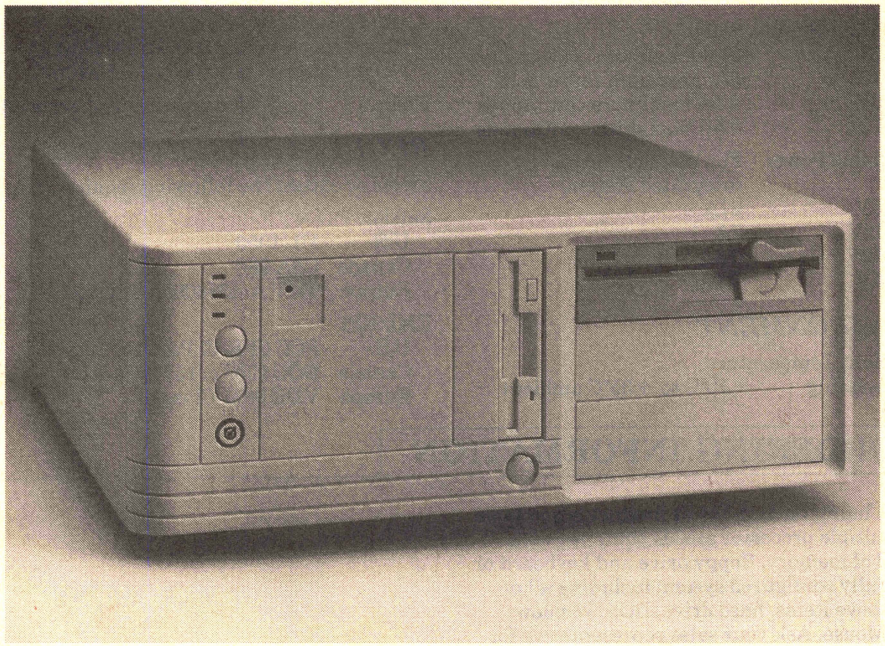
Safety	
U.S.	UL 1950
Canada	CSA 950
Europe	TUV to EN60950, IEC 950
EMI/RFI	
U.S.	FCC CFR 47 Part 15, Class B
Canada	DOC CRC c. 1374 Class B
Europe	VDE 0871 Class B

ORDERING INFORMATION

The Champion Mid-Tower product is available in multiple processor speeds equipped with 8 MB of memory, floppy drive and keyboard or as a fully configured system including all of the above items, hard drive, DOS, Windows, and Mouse. Ask your sales representative for exact configurations.

Contact your nearest Intel distributor.

i486™ CHAMPION EXPANDABLE DESKTOP PLATFORM



281578-1

i486™ EXPANDABLE DESKTOP PLATFORM WITH VESA LOCAL BUS

The Champion Series Expandable Desktop platform features affordable, flexible performance and is manufactured by Intel. The Expandable Desktop Champion is available in three powerful microprocessor speeds (i486™ SX25, i486 DX33 and i486 DX266). The system CPU is also upgradable using Intel's OverDrive™ processors. The Expandable Desktop is just that, offering six available ISA slots (including 1 VL slot), five peripheral expansion bays in a chassis designed for fast integration and easy servicing. The system is also equipped with a high-performance accelerated SVGA VESA local bus graphic subsystem. The Desktop Champion from Intel is the optimum solution for cost sensitive applications requiring the flexibility of an expandable PC platform.

FEATURES

- SX25/DX33/DX266 i486 Microprocessors and upgrades to Intel's current and future i486 OverDrive Processors
- 2 VESA local bus slots. One filled with VL SVGA Graphics Accelerator Card 1024 x 768 x 16 colors and 512 KB video DRAM standard (expandable to 2 MB video DRAM and 1280 x 1024 x 256 colors)

i486™ and OverDrive™ are trademarks of Intel Corporation.
*Novell is a registered trademark of Novell, Inc.

October 1993
Order Number: 281578-001

intel

i486™ CHAMPION EXPANDABLE DESKTOP PLATFORM

FEATURES

- 6 additional ISA expansion slots (5 open). 5 accessible drive bays, (3) 5.25" and (2) 3.5". Available in multiple configurations (including 4 MB and 8 MB memory options, hard drives, DOS, Windows, mouse and keyboards)
- 4 MB/8 MB On-board SIMM memory (Expandable to 128 MB)
- Optional secondary cache 32K, 64K, 128K, or 256K (Direct mapped write-back)
- Super I/O standard (1 parallel, 2 Serial, and keyboard). Floppy and IDE controller standard
- FCC Class B certified
- Novell* certified

BENEFITS

- Provides high performance today and room for increased performance tomorrow
- Local Bus enables VL add-in cards to fully utilize the processing power of the CPU. Excellent for graphics intensive applications, with upgrade capability. VL-bus disk controllers and network interfaces can reach optimal data transfer speeds
- Expansion capability provides tremendous flexibility for current and future configurations
- Order this box preconfigured and save time and money on integration
- Ideal for MS Windows environments with room for growth for future applications
- Allows CPU performance enhancements
- Saves time and expense of adding peripherals and communication devices
- Fully compliant for home or office
- Compatible in a Novell network environment

SPECIFICATIONS

CPU	i486 SX Microprocessor; 25 MHz i486 dX Microprocessor; 33 MHz and DX266 MHz
Performance Upgrade	Intel OverDrive processors; 50/66 MHz future Intel OverDrive processor for i486 DX266 (238-pin socket)
Graphic Subsystem	SVGA Add-in Card
VESA Add-in Card preinstalled	Cirrus 5426 Standard S3 805 Optional
Cache Option (Direct Mapped write-back)	32 KB/64 KB/128 KB/ 256 KB upgrade option using 20 ns SRAMs
On-board Memory Standard RAM	8 30-pin SIMM sockets 4 MB/8 MB (1 Mb x 9 SIMM)
Maximum RAM Upgrade RAM	128 MB (16 Mb x 9) 256 Kb x 9, 1 Mb x 9, 4 Mb x 9, 16 Mb x 9
Access Time	80 ns or faster
I/O Expansion Slots	8 ISA slots 2 VESA Local Bus/16-bit ISA slots 4 16-bit ISA slots 2 8-bit slots
BIOS Interface	AWARD EPROM
Integrated Super I/O Card (preinstalled)	
2 serial ports	Async, RS-232C, one 9-pin/one 25-pin
1 parallel port	Centronics compatible, 25-pin bi-directional
1 keyboard port	AT compatible
Floppy disk controller	1.44 MB on-board
IDE hard disk	Provided on I/O Card/ supports two hard drives
Board Dimensions	8.5" x 13", four layers
System Dimensions	
Height:	6.25"
Width:	16.35"
Depth:	16.75"
Weight:	21 lbs without peripherals

16

intel

SPECIFICATIONS

Drive Bays

External	3 5.25″ half-height accessible (1.6″)
Internal	2 3.5″ one-third height accessible (1.0″)

Floppy Disk (standard on every system)

Footprint	3.5″ one-third height (1.0″)
Capacity	1.44 MB high density

Accessories

Champion Series Desk Top Product Guide
Power cable, IDE hard drive cables, and miscellaneous H/W
Video Driver Diskette(s)

ELECTRICAL

AC Voltage/ Frequency	115V/60 Hz, 230V/50 Hz

DC Power Supply 200W

+5V	13.8A maximum continuous
+12V	6.30A maximum continuous
−5V	0.5A maximum continuous
−12V	0.5A maximum continuous

Available Power For ISA slots (in an integrated system)

+5V	19A
+12V	6A
−12V	0.3A
−5V	0.3A

ENVIRONMENT

Ambient Temperature

Operating	+10°C to +35°C ambient

Relative Humidity

Operating	85% @ 35°C
Non-operating	95% @ 35°C

Altitude

Operating	to 10,000 ft. (3048m)
Non-operating	to 50,000 ft. (15,240m)

Noise 42 dB with peripherals idle

REGULATIONS

Meets or exceeds the following requirements

Safety

U.S.	UL 1950
Canada	CSA 950

EMI/RFI

U.S.	FCC CFR 47 Part 15, Class B
Canada	DOC CRC c. 1374 Class B

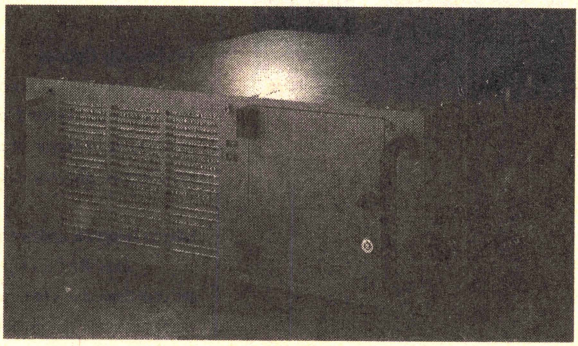
ORDERING INFORMATION

The Champion Desktop product is available in multiple processor speeds equipped either with 4 MB or 8 MB of memory, a floppy drive and keyboard or as a fully configured system including all of the above items, hard drive, DOS, Windows, and mouse.

Contact your nearest Intel distributor.

OmniRACK

The Universal Solution to Rackmountable Computing



Intel's new OmniRACK rackmountable microcomputer provides the ultimate in flexibility and value.

Designed to house multiple system architectures, OmniRACK provides a wide range of performance options, multiple upgrade paths, and a variety of price points.

OmniRACK is not just a rackmountable microcomputer system for industrial, communications and office automation applications; it is a complete environment that offers increased reliability, faster maintenance and easier integration.

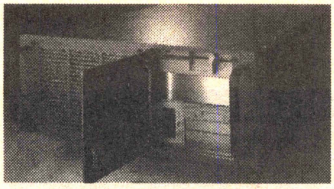
Features

- Supports Xpress and Baby AT architectures and 486™/SX25 through Pentium™ microprocessors.
- Pre-approved EMI/RFI and safety certifications, USA and international.
- Intel reliability and quality.
- High output 230W power supply.
- Filtered high performance cooling. 90CFM temperature controlled fans – single for Baby AT systems and dual for Xpress systems.
- Locking and latching door covering switches and drives.
- Quickly removable top cover and drive cage.
- Positive pressure drive bay – (2) 5.25" half-height; (1) 3.5" half-height - 1.6"; and (1) 3.5" third-height - 1".
- Optional card hold-down bracket.

Benefits

- Wide variety of performance and price points to fit different applications.
- Takes the legal worry out of supplying complete system solutions.
- High up-time.
- Extra +5V to support add-in boards.
- Reduces total energy usage while extending product life and minimizing maintenance costs.
- Selective security – lock or latch the door to prevent tampering or accidental system resets.
- Economical integration and rapid maintenance.
- Positive pressure drive bay forces airborne contaminants out of drives providing increased up-time.
- Additional shock and vibration protection.

October 1993
Order Number: 272400-001



Flexible Solutions

With OmniRACK, it is easy to tailor a solution to fit the application. The OmniRACK Classic E provides a low cost solution that can be upgraded as your application grows using Overdrive™ processors from Intel. OmniRACK Xpress/RX provides the ultimate in upgradability and modularity. The Xpress architecture protects your investment through several generations of microprocessors.

Two Security Options

Intel gives you two security options to protect the drives and the system reset, power and keyboard lock switches. With OmniRACK, you can lock the front panel door for maximum security, or simply latch it. Other rackmount chassis manufacturers give you only one option – locking.

Easy to Integrate and Service

OmniRACK's top access chassis design makes it easy to add controller and communications boards. The removable drive cage makes integrating and servicing drives a snap. In addition, Intel has already received EMI/RFI and safety agency certifications for OmniRACK, which takes the legal worry out of supplying system level solutions. To make your integration job easier, slide rails are available as an option. These rails allow you to mount OmniRACK in a standard 19" wide rack with a distance of 20" to 26" between the front of the rack and the rear mounting rail.

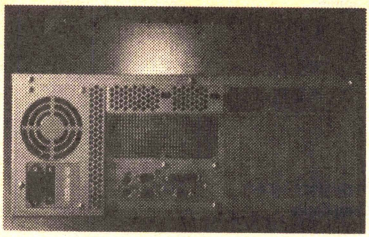
OmniRACK is available in standard configurations, or it can be purchased fully integrated to your specifications.

SPECIFICATIONS

Physical Dimensions	Length	17.8 in. (not including front panel)
	Width	17.0 in.
	Height	7.0 in.
Environmental	Temperature (operating)	0 to +50C
	Temperature (storage)	-25C to +70C
	Humidity (operating)	20% to 85% non-condensing
	Humidity (storage)	0% to 95% non-condensing
	Altitude (operating)	-300 to 10,000 feet MSL
	Altitude (storage)	40,000 feet MSL max.
	Vibration (operating)	Random Vibration
	5-350 HZ	.0001 G^2/Hz*
	350-500 HZ	-6 DB/octave rolloff from .0001 G^2/Hz (therefore at 500 Hz .00005 G^2/Hz)

* Note: G^2/Hz is the unit of measure of power or acceleration spectral density.

	Vibration (non-operating)	Random Vibration .001 G^2/Hz @ 5 Hz, sloping to .01 G^2/Hz @ 20 Hz, and .01 G^2/Hz @ 20 Hz to 1 KHz
	Shock (operating)	5 G, trapezodial, duration 11 ms
	Shock (non-operating)	40 G, trapezodial, duration 11 ms
Power Supply	Input Power	90 to 135 VAC or 180 to 265 VAC, switch selectable, 47 to 63 Hz



Output Power

Voltage	Minimum Continuous	Maximum Continuous	Peak Surge for 12 Seconds Max.
+5.00 VDC	3A	30A	–
+12.00 VDC	1A	6A	11A
-12.00 VDC	0A	.5A	–
-5.00 VDC	0A	.5A	–

(Total power output shall not exceed 230W continuous, 270W peak surge, 11 seconds maximum.)

Reliability

Table of MTBF Data

System Component	Calculated		Demonstrated	
	Test Conditions	Converted to 50° C	Test Conditions	Converted to 50° C
Classic E Baseboard	141,834 Hrs. @ 55°C	177,292 Hrs.	N/A - Test in progress	---
Xpress 6-Slot B/B	124,000 Hrs. @ 40°C	77,872 Hrs.	83,721 Hrs. @ 40°C	52,576 Hrs.
Xpress 8-Slot B/B w/o SCSI	88,000 Hrs. @ 40°C	55,264 Hrs.	28,414 Hrs. @ 40°C	24,831 Hrs.
230 Watt Power Supply	>100,000 Hrs. @ 25°C	29,600 Hrs.	28,414 Hrs. @ 40°C	17,831 Hrs.
Xpress CPU Modules				
Pentium 60 mHZ	123,000 Hrs. @ 55°C	153,750 Hrs.	19,582 Hrs. @ 40°C	12,297 Hrs.
X486DX 50	135,000 Hrs. @ 55°C	168,750 Hrs.	97,055 Hrs. @ 40°C	60,950 Hrs.
X486DX 266	218,000 Hrs. @ 55°C	272,500 Hrs.	17,831 Hrs. @ 40°C	11,197 Hrs.
X486DX 33	247,000 Hrs. @ 55°C	308,750 Hrs.	49,536 Hrs. @ 40°C	31,108 Hrs.
X486DX 25	247,000 Hrs. @ 55°C	308,750 Hrs.	44,064 Hrs. @ 40°C	27,672 Hrs.
Xpress memory module (w/8 mb total)	98,000 Hrs. @ 55°C	122,500 Hrs.	17,831 Hrs. @ 40°C	11,197 Hrs.

(These reliability statements provide information only, do not use contractually. Intel makes no warranty regarding the accuracy of these statistics.)

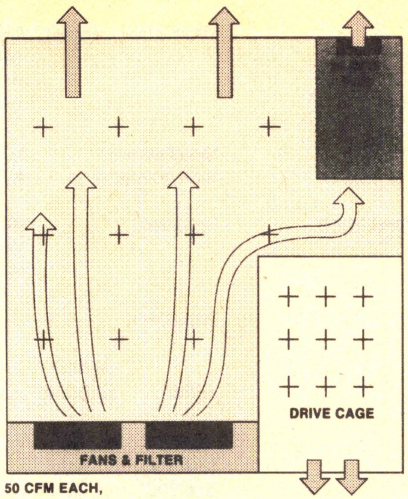
The prediction methodology used Intel, vendor or Bellcore failure rates.

Demonstrated MTBF values provided are limited by sample size and test time.

Demonstrated Field data collected on Intel OEM products predominantly show a much larger MTBF.

Demonstrated values are at a confidence limit of 80%.

Cooling	Front to back airflow	Air intake through front mounted filter (30 ppi). In addition to the power supply fan, 90CFM thermostatically controlled fans (single on Baby AT systems and dual on Xpress systems) provide positive pressure in the system.
Regulatory Agency Compliance	Safety	USA – UL listed – UL 1950, first edition Canada – CSA certified – CSA 22.2/950 Europe – GS license – EN 60 950 granted by TUV Rheinland of North America, Inc.
	EMI/RFI	Verified to comply with the following: USA, Canada – FCC-47 CFR, part 15, subpart B, class A Europe – VDE – class B 0871/6.78, in accordance with VFG 1991, and VFG 46,1992. – EN 55022 class B (which encompasses CISPR 22, class B) Japan – VCCI. class 2
Optional Slides for RETMA Rack Mounting	Chassis Trak, Slide Model No. C-300-S-1. Description: solid bearing, 3 piece, non-pivoting, cabinet mount slide, load capacity, 85 pounds.	



50 CFM EACH, UP TO 24° C, INCREASING TO 90 CFM @ OR ABOVE 35° C

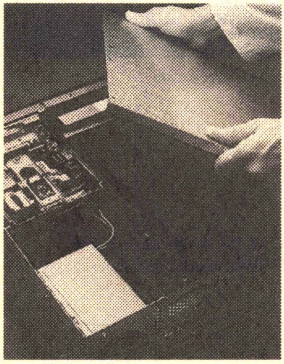
ORDER CODES

OmniRACK Classic E	OR486S257E4	7 ISA slot 486SX25 "Baby AT" form factor motherboard, base platform 4 MB memory std.
	OR486D337E4	7 ISA slot 486DX25 "Baby AT" form factor motherboard, base platform 4 MB memory std.
OmniRACK Xpress/RX	ORBASE6DT4	6 EISA slot Xpress architecture base platform, 4 MB memory std.
	ORBASE8LX8	8 EISA slot Xpress architecture base platform, 8 MB memory std.
Options	CARDHOLDBRKT	Add-in card hold down bracket
	ORSLIDERAILS	Slide mounting rails

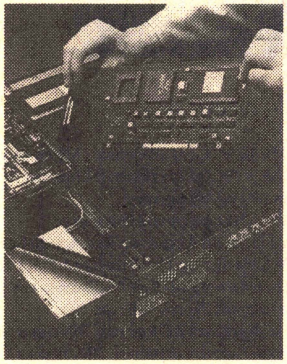
Serviceability

OmniRACK is designed for ease and speed of service and integration. The chassis is accessible from the top. All important functional components can be easily removed and replaced with a Phillips screwdriver. The four drive bays are easily accessible in a quickly removable drive cage.

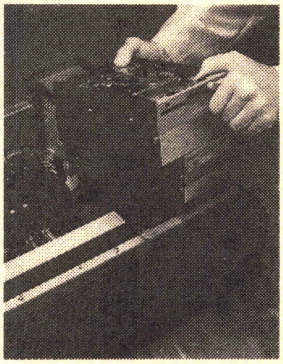
OmniRACK Xpress/RX typical maintenance times.



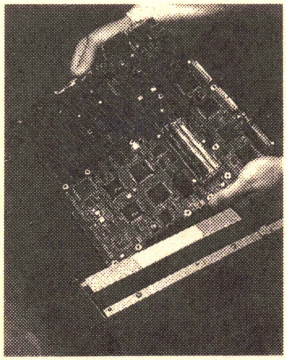
Remove top: slide from rack, remove four top screws, loosen screws on side and back:
45 seconds



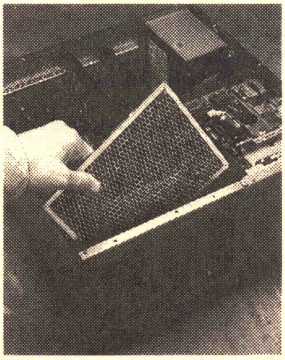
Remove CPU module (Xpress/RX):
20 seconds



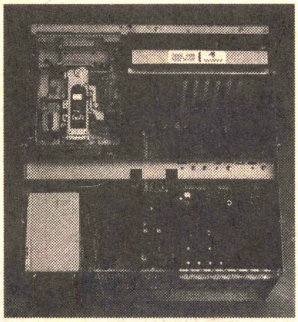
Disconnect cables and take drive cage out:
100 seconds



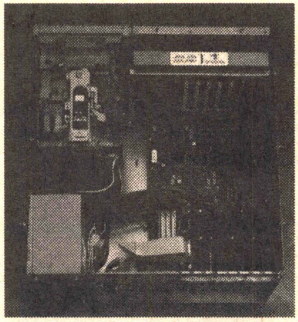
Remove base board:
120 seconds



Change filter:
30 seconds



Showing optional card hold down bracket.



Xpress architecture

"Baby AT" architecture

SYSTEM BOARD FEATURES

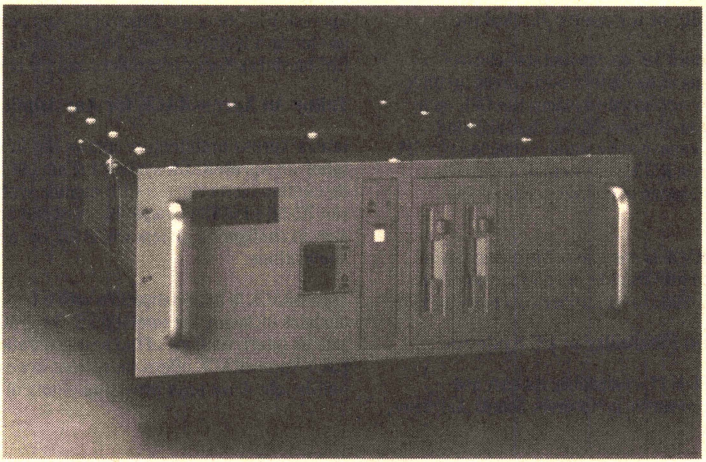
ORBASE6DT4 (6-slot Xpress architecture 4 MB memory standard)
ORBASE8LX8 (8-slot Xpress architecture 8 MB memory standard)

- Intel Xpress CPU module connector
- Intel Xpress RAM expansion module connector
- Six or eight full size EISA expansion slots capable of accepting PC/XT, ISA or EISA add-in boards
- Four SIMM RAM sites capable of accepting 1, 2, 4, 8, 16 or 32 MB SIMM modules, in pairs, for up to 128 MB
- Two RS232C serial ports with 9 pin DSUB connectors
- One parallel port with 25 pin DSUB connector
- PS/2-style keyboard interface (6 pin mini-DIN connector)
- PS/2 style mouse interface (6 pin mini-DIN connector)
- Integrated floppy disk drive controller
- Integrated IDE hard disk drive controller
- Integrated VGA controller with 512K upgradable to 1 MB video RAM, to support up to 1024 x 768, 256 colors

OR486S257E4 (7 ISA slot 486SX25 "Baby AT" form factor motherboard base platform, 4 MB memory standard)
OR486D337E4 (7 ISA slot 486DX33 "Baby AT" form factor motherboard base platform, 4 MB memory standard)

- Seven full size ISA (16 bit) expansion slots (includes a dual purpose VL slot)
- One VESA local bus slot
- Four SIMM RAM sites each capable of containing one 4M x 36-bit SIMM providing a maximum of 64MB
- Two RS232C serial ports with 9 pin DSUB connectors
- One parallel port with 25 pin DSUB connector
- AT-style keyboard interface
- Integrated floppy disk drive controller
- Integrated IDE hard disk drive controller
- OPTI chip set
- Optional second level direct-mapped, write-back cache using SRAM (64KB, 128 KB or 256KB)
- Phoenix BIOS in flash memory

Intel is a registered trademark of Intel Corporation. * Other brand and product names are trademarks of their respective owners.
All rights reserved. July, 1993 Order No.: 272400-001. Printed in U.S.A. HP/0793/6M/MR



A Reliable, Easy-To-Service Family Of Rack-Mount Microcomputers Featuring Modular Architecture.

Intel XpressRACK platforms are designed to offer a combination of the very best features of several system architectures.

There's the flexibility, expandability and upgradeability of passive-backplane systems combined with the cost efficiency, standardization and market acceptance of the PC.

These features make XpressRACK ideally suited for system integrators, OEMs and value added resellers seeking a versatile system that can be tailored to fit a wide range of applications and environments. From office automation servers to light industrial control, XpressRACK can meet your needs.

Built By Intel...for Reliability and Value

Because XpressRACK systems are built by Intel, you know they're reliable. The combination of superior design and high-quality components make XpressRACK a solid value.

FEATURES

- Xpress architecture with separate CPU module for easy upgrading. Accepts any Intel Xpress CPU module.
- Rugged, easy to service 19" RETMA rackmountable all-steel chassis.
- Six or eight industry-standard 32-bit EISA/ISA expansion slots.
- Four front-accessible, 5.25", half-height drive bays housed in positive air pressure enclosure.
- Up to 128 MB of on-board RAM (using 32-bit SIMMS). Up to 384 MB total RAM with optional memory expansion module.
- Baseboard includes controllers for super VGA, IDE hard-disk, and floppy disk.
- Two 50cfm fans, (with optional cartridge air filters), plus a power-supply fan.
- 396W auto-ranging power supply for eight-slot models. 300W switchable power supply for six-slot models.
- Parallel, dual serial, keyboard and mouse ports.
- Full one-year warranty – parts and labor.

October 1993
Order Number: 274136-003

The Upgradeability of a Passive Backplane

Built within a standard 19" rackmountable chassis, XpressRACK systems utilize Intel's own Xpress architecture. This innovative architecture places the CPU on a separate module from the baseboard, thus allowing performance to be enhanced by simply replacing the CPU module. So the XpressRACK can continue to grow in power as new generations of microprocessors are brought to market.

Expanding XpressRACK is easy, too. With either six or eight industry-standard EISA/ISA open expansion slots, it's easy to install additional controllers, memory and I/O.

The Economy and Flexibility of PC Systems

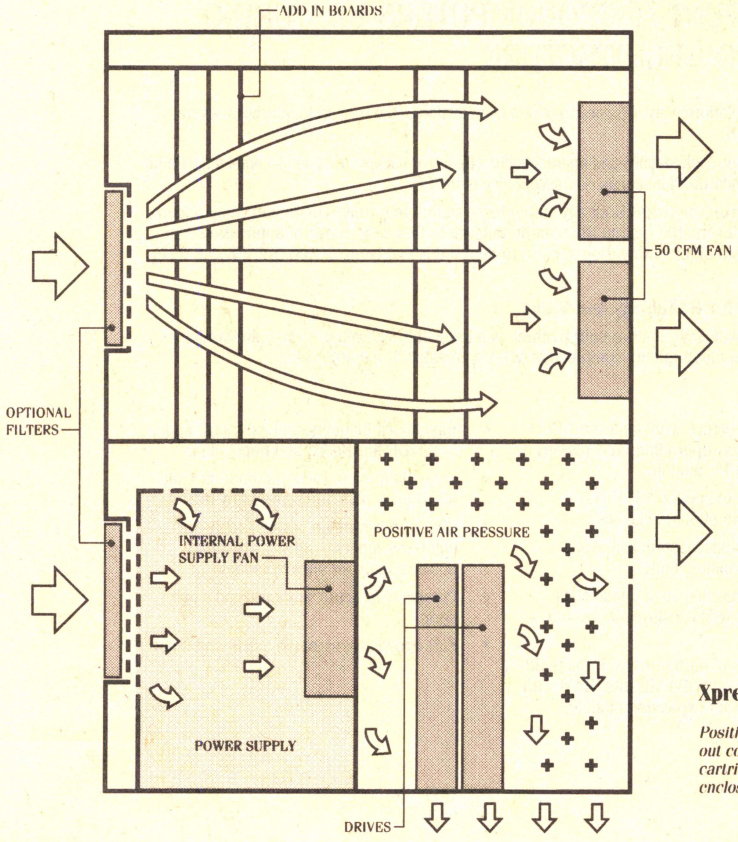
Because XpressRACK is PC-compatible, it's easy and economical to find compatible peripherals, boards and drives. XpressRACK offers a number of PC-compatible controllers as standard features. These include on-board super VGA, IDE hard-disk/floppy controllers and I/O ports.

Tailor An XpressRACK for Any Application

Intel's Xpress architecture allows the flexibility to truly tailor an XpressRACK system for almost any application. By combining an Xpress CPU module with a six or eight slot 32-bit EISA/ISA compatible card slot baseboard, it's easy to configure a system which fits the needs of most applications.

XpressRACK is fully compatible with all Intel Xpress CPU modules including the Intel486 25, 33, 2/50 and 50 MHz, as well as future versions of the next generation of Intel CPUs. Each XpressRACK offers four, front accessible 5.25" half-height drive bays providing plenty of room to grow.

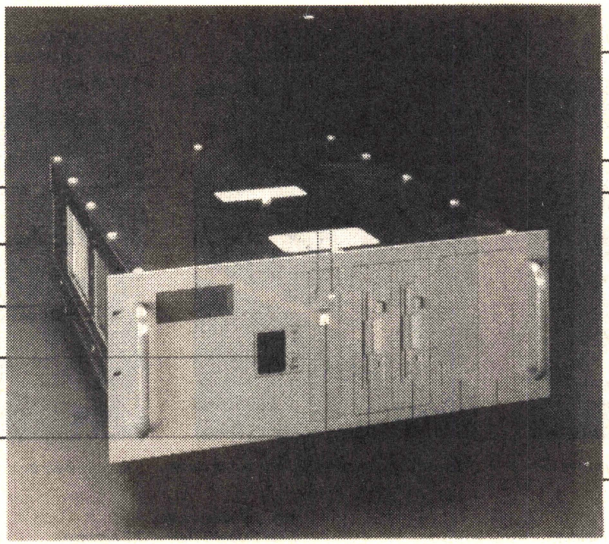
Superior Design



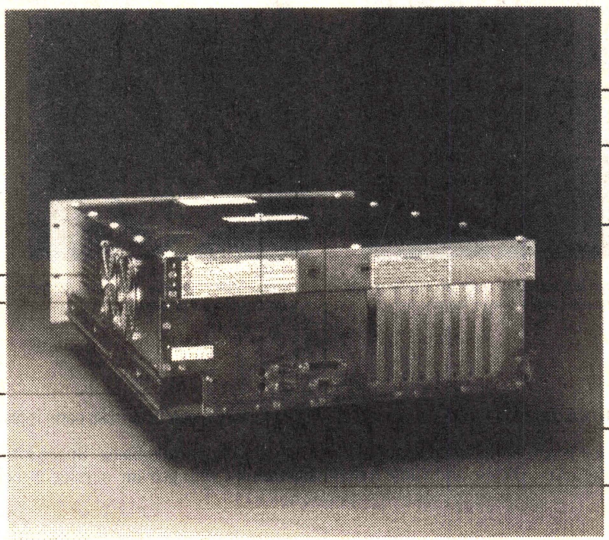
XpressRACK Air Flow

Positive air pressure in the drive bay keeps out contaminants, while optional air filter cartridges clean the air before it enters the enclosure.

Front



Back

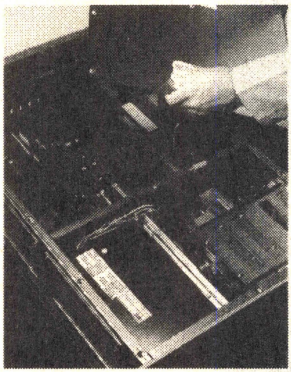


Unit does not ship with illustrated side rails.

SERVICEABILITY

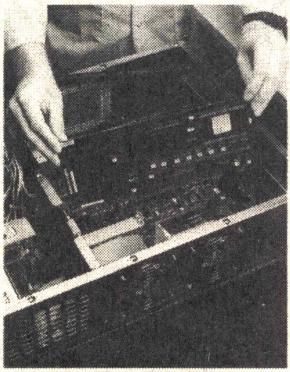
XpressRACK is designed for ease and speed of service. The chassis is accessible from the top. All important, functional components can be easily removed and replaced with a Phillips screwdriver. The four drive bays are accessible from both top and front.

Approximate Elapsed Time (AET): 30 seconds.



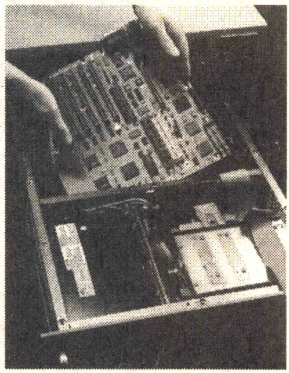
Remove top: slide from rack; quarter turn fasteners.

AET: 1.75 minutes or Total Elapsed Time of 2.25 minutes.



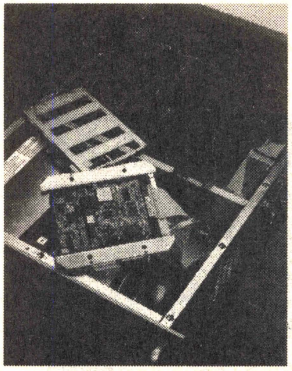
Remove CPU module and three add-in cards.

AET: 1.78 minutes or Total Elapsed Time of 4.03 minutes.



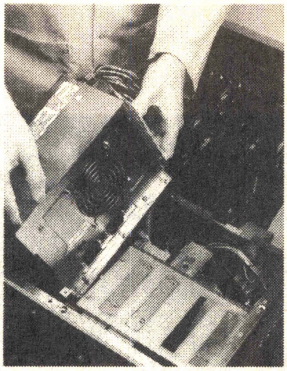
Remove base board.

AET: 1 minute.



Remove top and drive; note no interference with baseboard or power supply.

AET: 2.7 minutes or Total Elapsed Time of 3.7 minutes.



Remove power supply. Note no interference with baseboard or drives.

Warranty and Support

A full, one-year warranty offers a choice of carry-in depot repair, mail-in factory repair or Xpress parts replacement.

Additionally, on-site service is available by contract or on a time and materials basis. Spare kits and repair services are available for systems integrators, OEMs, and value added distributors who provide service direct to their customers.

SPECIFICATIONS

Baseboard

Baseboard	Xpress architecture

Memory

Baseboard	4 SIMM sites accept 1, 2, 4, 8, 16 or 32 MB SIMMS. Combine in pairs to provide up to 128MB.
RAM expansion module	256MB additional RAM

I/O

2 serial ports	Asynch RS-232-C, 9 pin AT-compatible
1 parallel port	Centronics compatible, 25 pin AT-compatible
1 keyboard port	PS/2 style connector
1 mouse port	PS/2 style connector
1 video port	Integrated super VGA with 512K upgradeable to 1MB
1 IDE/floppy	Integrated controllers

Expansion Slots

6 slot base	6-16-bit ISA/32-bit EISA
8 slot base	8-16-bit ISA/32-bit EISA

CPU Modules and Memory Cards

Any Xpress CPU module or memory card

Peripheral Bays

4 front access 5.25" half-height, exposed or concealed

Regulatory

Safety	
US	UL Listing UL 478 4th edition
Canada	CSA Certified C22.2 No 950
Europe	GS License EN 60950
EMI/RFI Compliance	
US/Canada	FCC 47CFR Part 15, Subpart J, Class A
Europe	CISPR 22 Class B VDE 0871/6.70 Class B
Japan	VCCI, Class 2

Enclosure

Dimensions	19"W x 7"H (4 RETMA units) x 25.6"D
Weight	42 lbs.
Mounting	Threaded inserts for General Devices Chassis Trak series C300 slides
Construction	All steel parts are plated for corrosion control

Humidity (non condensing)

Operating	20 to 85%
Non-Operating	0 to 95%

Ambient Temperature

Operating	5°C to 50°C
Non-Operating	-25°C to 70°C

Elevation

Operating	10,000 ft., max
Non-Operating	40,000 ft., max

Vibration

Operating	.001G/Hz @ 5Hz, sloping to .01G/Hz @ 20Hz, .01G/Hz @ 20Hz to 1000Hz
Non-Operating	30G, 1/2 sine wave, with 11msec duration

Cooling

2 50CFM fans with optional cartridge filters provide cross flow ventilation of baseboard, CPU, RAM, and expansion boards.

Power supply fan provides positive pressure cooling in drive bays.

Electrical

PC form factor power supplies

6 slot base unit 300W power supply

Input power: 90 to 132 VAC or 180 to 264 VAC switch selectable, 47 to 63 Hz.

Output power

Voltage	Minimum Continuous	Maximum Continuous	Peak Surge for 15 Seconds Max
+ 5.00 VDC	7A	35A	—
+12.00 VDC	.5A	10A	14A
- 12.00 VDC	0A	.5A	—
- 5.00 VDC	0A	.5A	—

Total power output shall not exceed 300W continuous

8 slot base unit 396W power supply

Input power: 90 to 132 VAC or 180 to 264 VAC automatic range switching, 47 to 63 Hz.

Output power

Voltage	Minimum Continuous	Maximum Continuous	Peak Surge for 15 Seconds Max
+5.00 VDC	11A	45A	—
+12.00 VDC	0A	13A	19A
- 12.00 VDC	0A	1A	—
- 5.00 VDC	0A	.5A	—

Total power output shall not exceed 396W continuous

ORDERING INFORMATION

Base Platform:

Consists of:

Chassis, power supply, peripheral enclosure, plus a choice of:
6 slot Xpress baseboard with 4MB of memory, or
8 slot Xpress baseboard with 8MB of memory

Order Code	Number of Slots	Power Supply
RXBASE6E4P1	6	300W
RXBASE8E8P2	8	396W

Options:

Order Code	Feature
XPRMFILTER	Set of removable cartridge air filters

Xpress Intel486™ CPU Modules

Order Code	Description
*BXCPU486DX266	Intel486DX2/66 CPU module
BXCPU48650	Intel486DX50 module
BXCPU486DX250N	Intel486DX2/50 module
BXCPU48633	Intel486DX33 module
BXCPU486SX25	Intel486SX25 module

*Available after September 1, 1992

Intel Sales Offices

For more information or the number of your nearest sales office, call 1-800-548-4725.

United States
Intel Corporation
2200 Mission College Boulevard
P.O. Box 58119
Santa Clara, CA 95052-8119

Japan
Intel Japan K.K.
5-6 Tokodai, Tsukuba-ski
Ibaraki, 300-26

France
Intel Corporation S.A.R.L.
1, Rue Edison, BP 303
78054 Saint-Quentin-en-Yvelines Cedex

United Kingdom
Intel Corporation (U.K.) Ltd.
Pipers Way – Swindon
Wiltshire, England SN3 IRJ

Germany
Intel GmbH
Dornacher Strasse 1
8016 Feldkirchen bei Muenchen

Hong Kong
Intel Semiconductor Ltd.
10/F East Tower – Bond Center
Queensway, Central

Canada
Intel Semiconductor of Canada, Ltd.
190 Attwell Drive, Suite 500
Rexdale, OH M9W 6H8

Information contained herein supersedes previously published specifications on these products from Intel and is subject to change without notice. i486™, i386™ and SX™ are trademarks of Intel Corp. Intel is a registered trademark of the Intel Corporation and all other trademarks are owned by their respected companies.

Support Products 17

iRUG DESCRIPTION

iRUG is the Intel Real-Time User's Group. It is an incorporation chartered to establish a forum for users of the iRMX and iRMK operating systems and to promote and encourage development of Intel based software.

Benefits of membership include: access to the user's library of software tools and utilities; membership in local and international chapters; access to the group bulletin board; receipt of quarterly international magazines; opportunity to present papers and conduct workshops; invitations to seminars devoted to the use of Intel products.

The user's library, maintained by iRUG, contains software programs written and submitted by members and Intel employees. Programs available range from file or directory manipulation commands and terminal attribute selection utilities to dynamic logon, background job facilities and basic communication utilities.

Local and international iRUG chapters provide a forum for members to meet other Intel Real-Time system users in an informal setting. At local meetings and the annual international seminar, members can discuss their ideas, share their experiences and techniques, and give feedback to Intel for future improvements and features of Intel products and those supplied by other companies.

The "Intel Forum" sponsors iRUG Special Interest Group (SIG) on the CompuServe Information Service. The message facility (bulletin board) allows members to leave and receive messages from other members. These might include problems and solutions regarding the iRMX operating system or new techniques to be shared.

"The Human Interface" is iRUG's quarterly international magazine. It serves as a supplement to chapter meetings by providing: library listings, information on the latest releases of products; officer messages; member SPRs; vendor ads; release and update plans; and technical articles from the members.

For Information Contact:

iRUG
P.O. Box 91130
Portland OR 97291
1-800-255-iRUG

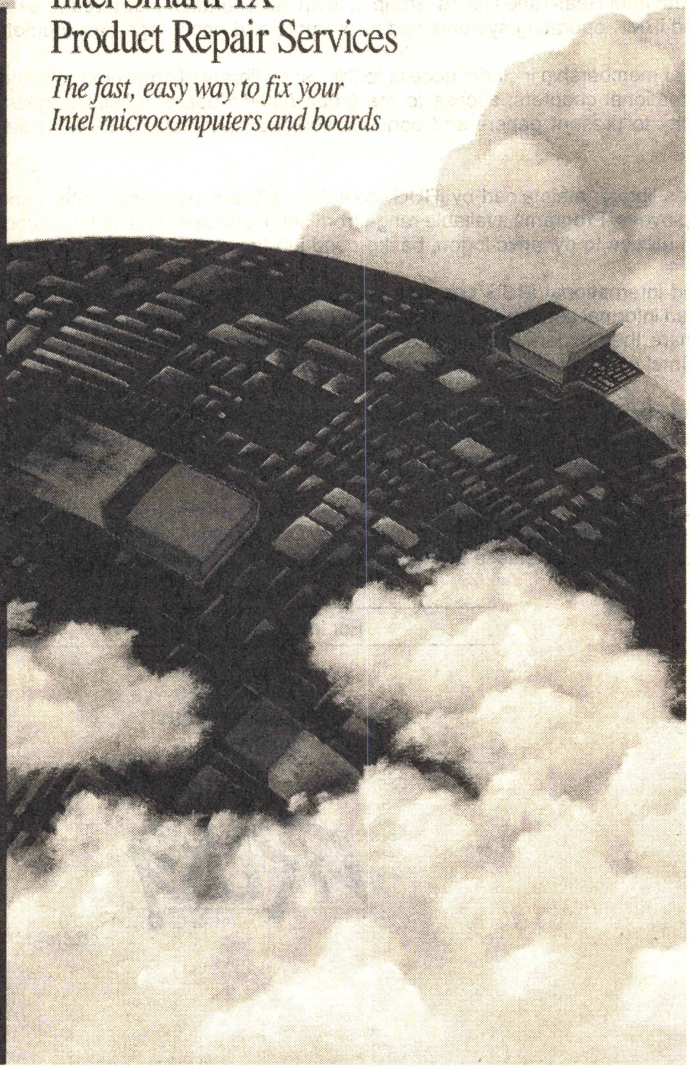


280678-1

17

Intel SmartFIX Product Repair Services

The fast, easy way to fix your Intel microcomputers and boards



intel

October 1993
Order Number: 274163-001

Despair over product repairs no longer. Intel's SmartFIX repair services are the fast, easy way to get Intel products repaired and back in service in record time, with record ease. SmartFIX includes a broad selection of services that meets the repair needs of original equipment manufactures (OEMs), custom solution providers, distributors and their customers (VARs) during the entire time they own their product.

SmartFIX is a new, innovative method for accelerating product problem resolutions. When you have a mail-in repair need, just use our SmartFIX Fast and ExtendIt repair packages. They're damage-resistant and easy to use. Simple slip the faulty part inside and mail it to us. We repair it in five days or less and rush it back. You get SmartFIX repairs where you buy your Intel products. Now, more the ever, quality service is as easy as A, B, C!

The range of SmartFIX services includes five-day repairs on subassemblies, an overnight emergency replacement parts service, flexible extended warranties, ten-day repairs on complete systems, and no-questions-asked platform exchanges (within the first 30 days). We even provide volume repairs tailored for OEMs building system boards.

No matter which Intel board or system products you own, there's a SmartFIX repair solution that makes life easier for you.

Intel Repair Centers:

There When You Need Us, Where You Need Us

Features

20,000 sq. ft. Intel quality-certified repair center.
State-of-the-art GenRad and Zehntel automated test equipment.
Functional at-speed test fixtures (FASTs) for complete interoperability and voltage margin analysis.
Damage-resistant shipping packages.
Products updated to latest manufacturing revision level.
All parts and labor necessary for complete repair.
Board cleaning in a non-CFC cleaning agent for a safer environment.

Benefits

Investment protection through the product's entire life cycle.
Super-easy to use.
Fast turnaround on repairs.
Reduced maintenance and inventory costs.
Convenience of getting repair solution where you bought your product.
Predictable repair costs.
Twelve-month warranty on out-of-warranty repairs.

Intel repair centers keep Intel products working all the time, all over the world. They're the only place you'll find Intel-trained architecture experts repairing products using the world's most advanced electronic test equipment. Not only our knowledge base but our cycle times are unparalleled in the industry. Our speedy turn-times get failed parts back into circulation quickly, saving money by reducing replacement inventory. No one else has the ability to: repair your part to the latest Intel manufacturing level, rush Intel factory parts to you overnight, or be there for you anywhere in the world, with the Intel service solutions you need.

Three Worldwide Locations

Currently we have repair centers in three locations: Arizona, England and Japan. Each center is a self-sufficient operation with test engineers, quality control engineers, repair technicians, production control staff, assemblers and other experts on staff. Rigorous training at all levels keeps our personnel current on Intel technology as well as on state-of-the-art test equipment and methods.

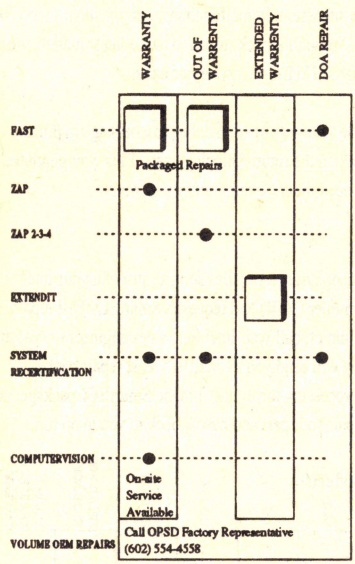
State-of-the-Art Equipment

Our equipment is the latest in electronic test gear and includes GenRad* and Zehntel* in-circuit testers as well as sophisticated board and system-level testers. Every board and system that runs through our centers receives comprehensive voltage, temperature and timing tests, and leaves with a complete failure analysis report to help us — and you — improve future products and processes.

Better Than New

Every board you send in is repaired to the latest Intel manufacturing revision level (unless you specify otherwise). That means the part you get back has performance and quality similar to that of a newly manufactured part, at no extra cost.

SmartFIX Repair Programs



Intel Quality in Every Repair

Your repairs are protected by Intel quality and reliability every step of the way. Special Fast and ExtendIt shipping packages carry your part to us in a static-free nickel bag with shock protection padding. Once in our Intel quality certified repair center, your part is handled gently and professionally to protect it from electro-static discharge (ESD) damage. Repair center employees wear ESD-resistant smocks and grounding straps, and our floors are ESD-treated. Once repaired, we ship your better-than-new part back to you in another static-free, padded package.

Minute-by-Minute Accountability

Our product tracking system is as sophisticated as our test equipment. Intel's automated tracking system follows every product through the repair process from beginning to end. When your product enters our center, we assign it a control number that locates it at any time. When the repair is completed, we provide you with a comprehensive written analysis of the repair, including a component failure analysis. This data helps us improve our design and manufacturing processes to eliminate future problems and provide you with the highest quality product you can buy.

Continuous Improvement

Intel repair centers are linked to Intel manufacturing for continuous two-way communication. Manufacturing notifies us of all engineering change orders, which we implement in our repairs. Our repair centers, in turn, provide manufacturing with feedback that helps them improve future products.

We Stand By Our Work

We warranty every out-of-warranty repair for a full year, evidence of our confidence that Intel provides quality repairs that keep on working for a long time to come.

SmartFIX Fast Repairs

For In and Out-of-Warranty Products

Qualifications

Available to in-warranty and out-of-warranty customers.

Continental U.S.

Proof of warranty required for free repair.

Intel-manufactured products only.

Supports PC boards and platforms, multibus boards and systems, and in-circuit emulators.



The Easiest Way to Repair Your Intel Products

SmartFIX Fast repairs are the most popular way to repair Intel boards and systems. You get SmartFIX Fast shipping packages from Intel just as you would any other Intel product. Then, when a subassembly malfunctions, simply slip it in the Fast package and mail it to Intel. We send it back, repaired to the latest manufacturing revision level, within five days.* No hassle, no service calls.

With Fast repairs, we provide a damage-resistant shipping package to protect your investment; we also pick up the return shipping costs. Plus, you purchase Fast repairs only when you need them, saving money and worry.

How to Order and Use

When one of your Intel parts or systems requires service, just call your Intel distributor or value-added reseller (VAR) and request a SmartFIX Fast repair. (If you are an OEM buying direct, call your Intel field sales engineer or customer service representative.) They will send you a SmartFIX Fast repair package. Simply complete the brief processing form, pop it in the SmartFIX package, and drop it in the mail. We'll repair your part and have it back to you in no time.

SmartFIX Supported Products

	FAST	ZAP	ZAP 2-3-4	EXTENDIT	FIELD DEPORT EXCHANGE	VOLUME OEM REPAIRS "Intel Inside"
MICROCOMPUTER PRODUCTS						
Platforms	●			●	●	Volume OEM board repair for Intel component customers building boards
Motherboards	●	●	●	●	●	
Memory Boards	●	●	●	●	●	
CPU Boards	●	●	●		●	
Power Supplies	●	●	●		●	
MULTIBUS I & II						
Platforms	●				●	
CPU Boards	●				●	
Memory & Add-In Boards	●				●	
EMULATORS						
Development Tools (ICE Products)	●				●	
NETWORKING						
Network Products (NetSight & Storage Express)	●				●	
MULTIMEDIA						
DVI Action Media	●					

If you don't know who to call for repairs, call Intel at 1-800-628-8686

**Not including transit time. Allow 10 days for platforms.*

SmartFIX ZAP Repairs

For Major PC Subassemblies

Qualifications

Advanced customer account setup required prior to first order (72 hours required if first order includes setup).
Problem isolation performed by customer.
Only for major replaceable PC subassemblies.
ZAP available free for parts under warranty.
ZAP 2-3-4 available on a fee basis for out-of-warranty products.
Continental U.S. shipments only.
Intel-manufactured products only.
Subject to parts availability.
Peripherals added by distributors not covered by ZAP.

Supported Products

Motherboards
Power supplies
Memory boards
Xpress baseboards
Xpress CPU modules

When You Absolutely, Positively Have to Have It the Next Day

Zoom-a-Part (ZAP) is Intel's emergency parts exchange service that rushes a replacement part to you within 24 hours* of your call. ZAP service is available for all major replaceable PC subassembly parts (not platforms) in and out of warranty.

We carry a complete parts inventory to respond to your needs. If you happen to have an older product that needs repair, we will ZAP it to you at exactly the same revision level.

Use ZAP parts exchange when you absolutely, positively have to have a part the next day to meet a critical repair need.

How to Order and Use

Before you can get your first ZAP, you have to have an account set up. Just call your Intel customer service representative and request that a ZAP account be opened for you. They'll ask you to send a P.O. number, your business address and billing address. Then you're set. If you haven't set up a ZAP account in advance of your first call, the ZAP request will take an extra 48 hours.

Direct customers and VARs simply phone their ZAP order into their Intel distributors, and they'll have a new part express-mailed to you the next business day. If you're a direct OEM, call your customer service representative or 1-800-628-8686 for assistance. Meanwhile, box up the faulty part and return it to us within seven days.

If you don't know who to call for repairs, call Intel at 1-800-628-8686

**Subject to parts availability.*

SmartFIX ExtendIt Repairs

Flexible Warranty Extension

Qualifications

Available within the first 60 days of ownership.

Minimum purchase: 10 ExtendIt warranties per order of the same product.

Continental U.S. only.

Problem isolation performed by customer.

Proof of warranty extension and completed processing form required.

Extended Warranty Repairs Over the Life of Your Product

SmartFIX ExtendIt is Intel's new warranty extension program for PC platforms and PC motherboards. Purchase an ExtendIt flexible warranty extension from your distributor or VAR any time during the first 60 days of product ownership. Then, if anything goes wrong, simply return the failed part to Intel in your ExtendIt package for repair. ExtendIt warranties are an economical, hassle-free service solution for OEMs and custom solution providers who want to keep their Intel products working non-stop.

ExtendIt warranties cover PC platforms and PC motherboards for up to 4 years beyond the factory warranty. As with SmartFIX Fast repairs, ExtendIt repairs are out of our factory within 5 days and shipped back to you.

How to Order and Use

To order ExtendIt warranties, call your Intel distributor or VAR during the first 60 days of product ownership. Tell them you want to purchase SmartFIX ExtendIt warranties. (If you are an OEM buying direct, call your Intel field sales engineer or customer service representative.) Your distributor will send you a SmartFIX ExtendIt warranty repair package.

When a part needs repair, simply fill out the processing form, slip the part in the ExtendIt mailing package and drop it in the mail. No phone calls, no hassles. Your part will be repaired within five days of receipt and returned to you with a new package for future use.

If you don't know who to call for repairs, call Intel at 1-800-628-8686

SmartFIX for OEMs

Get Your OEM Out of the Repair Business

Your Competitive Edge

SmartFIX gives OEMs a competitive advantage by out-sourcing the expensive repair operation. Reduce overhead and turn inventory faster (not to mention, use less of it!), all while continuing to focus on your main line of business. We'll customize a SmartFIX program for you, meeting your repair needs the way you want them met.

Original equipment manufacturers (OEMs) don't like repair hassles any more than their customers. In fact, repair-related productivity losses and the costs of excess inventory, production interruptions and capital equipment are of great concern to the volume manufacturer.

Repairs for Boards Based on Intel Architecture

That's why we offer SmartFIX for OEMs, an easy, cost-effective repair program for volume manufacturers of Intel architecture-based boards and systems. It works just like our original SmartFIX program. But instead of sending in individual boards and systems for repair, OEMs can send in dozens at once. The turnaround is just as fast, the process just as easy.

Ship All Your Problems to Us

OEMs ship faulty boards and systems back to Intel weekly, monthly or at any desired frequency in special SmartFIX Fast containers for multiple boards. These crates hold several dozen boards, so you can easily get batches of faulty boards out of your facility and into ours for fast, flawless repair.

Focus on What You Do Best

SmartFIX for OEMs gets OEMs out of the time-consuming and cost-intensive repair business. By moving repairs off-line to Intel's repair experts, SmartFIX lets OEMs concentrate on what they do best: manufacture products. You don't have to stop the line to troubleshoot problem boards, invest in expensive test equipment and staff, and keep large stocks of inventory on hand. SmartFIX for OEMs moves the entire repair operation off-site, reducing overhead and increasing your bottom line.

Select the Process That Meets Your Needs

SmartFIX for OEMs has a flexible menu of repair options. Cycle times range from two to five days (plus transit time), so your inventory isn't tied up in repair limbo for long periods. Instead, we fix faulty parts, putting them back into circulation quickly, where they make money for you. Additional service selections include choice of type and depth of testing, failure analysis and reporting, and complex failure analysis and repair.

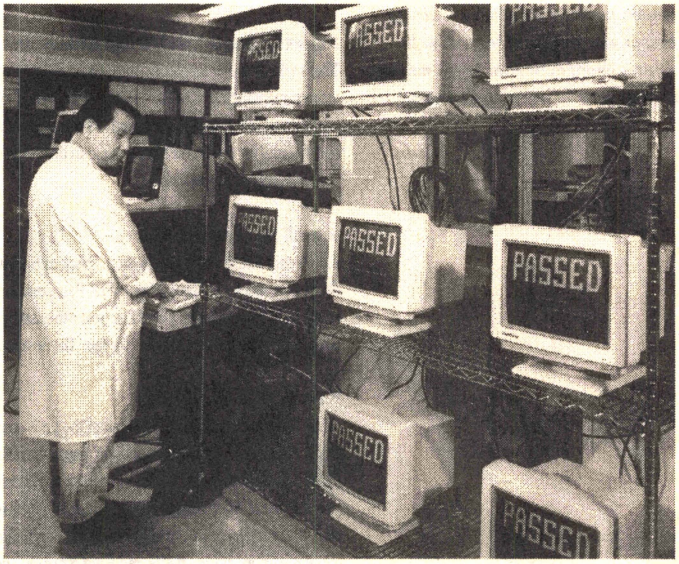
SmartFIX for OEMs eliminates the cost of maintaining your own repair center. You get the most knowledgeable support possible for Intel products — Intel — repaired with state-of-the-art test equipment and techniques.

If you're an OEM who wants to get out of the repair business and stay in the manufacturing business, ask about SmartFIX for OEMs. And let Intel fix it for you.

Specialized Product Center

Your Partner In Custom Manufacturing And Complex Systems Integration

SPC focuses on opportunities that utilize our expertise in three major areas: (1) complex integration, (2) custom systems, and (3) custom board and system software design.



Intel's Specialized Product Center (SPC) is a complete, fully staffed integration center that designs, assembles and manufactures custom systems to meet the unique needs of distributors and OEMs in the high-tech industry. Partnering with the SPC allows customers to take advantage of new business opportunities that they would otherwise not be able to pursue because of constrained resources or lack of complex systems expertise. The benefits to our customers include fast turnaround, access to the latest Intel technology, complete documentation and support, and reduced time-to-market.

intel

October 1993
Order Number: 274164-001

Intel's expertise, technology and integration capabilities help our customers expand their markets without adding to their overhead.

By utilizing the integration expertise and resources of Intel's Specialized Product Center (SPC), the companies we partner with are able to extend their capabilities and expand their markets repidly without increasing their overhead expenses or adding unnecessary support costs.

The Specialized Product Center gives our customers a capability not usually found in large companies -- flexibility. SPC can turn highly specialized configurations in five days or less, helping our OEMs and distributors keep their competitive edge and provide better service to their customers.

SPC's capabilities are especially useful for companies who serve value-added, niche markets that require custom chassis or features such as multimedia.

SPC focuses on opportunities that utilize our expertise in three major areas: (1) complex integration, (2) custom systems, and (3) custom board and system software design.

SPC provides:	Partner benefits:
Multiple cards and peripherals	Flexibility to meet your exact needs and specifications
Custom software configuration	Fast turnaround
Hardware and software testing	No additional overhead
Custom chassis design based on Intel technology	No capital investment
EMI, RFI, safety, and burn-in	No drain on personnel resources
Custom board design	Access to the latest Intel technology
System software design	Agency tested and certified
Engineering consulting and project management	Complete documentation and support
Access to new Intel technology	Faster time-to-market

Custom systems to meet unique needs.

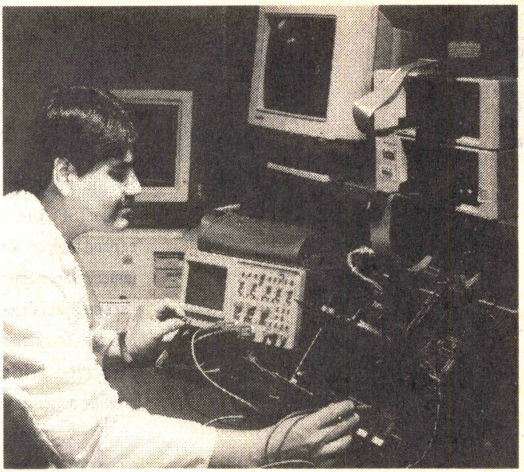
The computer systems market is changing. An enormous investment of time, expertise, manufacturing resources, and capital are required just to stay competitive.

Intel's Specialized Product Center gives customers the capability to not just keep up with the market, but to stay ahead of it.

We specialize in providing custom systems to meet the unique needs of our partners and we're committed to delivering solutions that offer the flexibility, expandability and upgradeability of Intel Product Group (IPG) product lines.

XpressRACK, for example, fills a need in the market for a rack-mountable microcomputer that has the flexibility of EISA/ISA architecture and the upgradeability of a passive backplane.

The SPC took Intel's upgradeable Xpress architecture and put it in a custom chassis. And we did it fast.



Here's what some of our partners are saying about SPC:

"You did what we asked and I got my order in five days."

"I can't believe a big company is so flexible!"

"I won the business because I knew I could count on SPC to deliver my product on time."

How do you partner with SPC?

One of the elements that has made the SPC successful is putting a process into place that helps our partners use us quickly and easily.

Here's how the process works:

1. Locate quick order custom configurations in the Price Book and order through a CSR. Configurations selected from the Price Book are delivered within five working days.
2. Contact SPC Business Development Managers (BDMs) for complex integration of custom designs (602-554-4558).

Partnering with the SPC can help you build stronger relationships with your customers.

In today's highly competitive market, the key to holding onto your customers is being able to satisfy all of their needs, quickly, and with a quality product.

So the next time one of your customers comes to you with an application that involves complex integration or custom systems design, don't let the opportunity, or the customer, get away.

Contact Intel's Specialized Product Center.

We'll put our capabilities to work...for you.

INTEL SYSTEMS ENGINEERING SERVICES



274137-1

BRING YOUR PRODUCT TO MARKET FASTER FOR GREATER PROFITS

Your project deadlines are tight. You have more to do in less time, and yet, still more is expected of you. Increase your product quality. Cut your manufacturing costs. Deliver your product to market quick. In this fast-paced world, you need an edge on the competition.

Let Intel's Systems Engineering Services provide you that competitive advantage. We'll help you design, develop and manufacture your Intel architecture-based products with the high quality standards and in the timely manner you've come to expect from us.

Intel Systems Engineers (SEs) know Intel microprocessors, boards, systems, and software better than anyone else. They are part of the Intel engineering team. By tapping in to the vast experience, expertise, and resources of Intel, you'll dramatically reduce your time to market, accomplishing in weeks what might have taken months to do alone.

17

INTEL SYSTEMS ENGINEERING SERVICES

- Hardware Design
- Software Design
- Project Management
- Design Validation
- Systems Analysis and Debug
- Network Design
- Systems Administration
- Board Design and Manufacturing

iRMX® and Intel are registered trademarks of Intel Corporation.
UNIX is a registered trademark of AT&T Bell Laboratories.
DOS and Windows are registered trademarks of Microsoft Corporation.
OS/2 is a trademark of International Business Machines Corporation.

September 1993
Order Number: 274137-002

intel.

WHAT YOU GET FROM AN INTEL SYSTEMS ENGINEER

EXPERIENCE

The average Intel SE has over a decade of experience working on real-world projects. That experience allows them to resolve issues quickly, complete projects on time and incorporate current technological developments to bring your product to market early and within budget.

EDUCATION

Intel SEs typically possess Electrical Engineering and/or Computer Science degrees. And Intel's continuing education program means SEs are trained on and understand the latest advances in today's technology.

EXPERTISE

With experience working with both Intel and non-Intel components, systems, software and networks. SEs can offer the most appropriate resources to create a product that meets **your** requirements.

BREADTH

Each Intel SE works an average of six to eight projects every year. As a group, Systems Engineering completes more than 300 projects each year. Bring in an SE that's right for your project, with the specific experience and knowledge necessary for on-time completion and success.

FLEXIBLE RELATIONSHIPS

Intel offers the right contract flexibility to meet your specific project needs. Short term or long term. Weekly or monthly, Intel SEs are available on a fee basis that's designed to fit your product development plan's budget and schedule.

INTEL SYSTEM ENGINEERING SERVICES

SOFTWARE CONSULTING

Intel offers specialized engineers with relevant experience in working with industry standard operating systems, including DOS*, OS/2*, Windows*, iRMX* and UNIX*.

These SEs have direct access to Intel's source code and design tools. Such resources enable Intel SEs to quickly and cost-efficiently design and develop customized software for your target systems.

DESIGN CONSULTING

The beginning stages of any project are critical to its success. Minor design mistakes can negatively impact an entire project. Intel's SEs provide the experience and expertise you need to develop the best design without the flaws that can lead to cost overruns and time delays.

INTEL FOLLOW-ON SERVICE

BOARD MANUFACTURING

Not only can Intel manage your project and design your board, Intel also offers world-class manufacturing to complete the cycle.

All custom boards are manufactured in strict accordance with Intel's own production, quality assurance and test procedures. These procedures conform to international ISO9000 standard, as well as many national standards.

And by using the latest technological advancements in manufacturing, like just-in-time production methods, Intel produces the highest level product at a low, competitive cost.

CALL IN THE EXPERTS ... CALL INTEL

The next time you're asked to get a complex design project off and running quickly and economically, call the Intel Systems Engineering Services Manager nearest you. Other office listings may be obtained by calling 800-548-4725 (U.S./Canada)

NORTH AMERICAN SALES OFFICES

ALABAMA

Intel Corp.
600 Boulevard South
Suite 104-I
Huntsville 35802
Tel: (800) 628-8686
FAX: (205) 883-3511

ARIZONA

†Intel Corp.
410 North 44th Street
Suite 500
Phoenix 85008
Tel: (800) 628-8686
FAX: (602) 244-0446

CALIFORNIA

Intel Corp.
3550 Watt Avenue
Suite 140
Sacramento 95821
Tel: (800) 628-8686
FAX: (916) 488-1473

†Intel Corp.
9655 Granite Ridge Dr.
3rd Floor, Suite 4A
San Diego 92123
Tel: (800) 628-8686
FAX: (619) 467-2460

Intel Corp.
1781 Fox Drive
San Jose 95131
Tel: (800) 628-8686
FAX: (408) 441-9540

*†Intel Corp.
1551 N. Tustin Avenue
Suite 800
Santa Ana 92701
Tel: (800) 628-8686
TWX: 910-595-1114
FAX: (714) 541-9157

†Intel Corp.
15260 Ventura Boulevard
Suite 360
Sherman Oaks 91403
Tel: (800) 628-8686
FAX: (818) 995-6624

COLORADO

*†Intel Corp.
600 S. Cherry St.
Suite 700
Denver 80222
Tel: (800) 628-8686
TWX: 910-931-2289
FAX: (303) 322-8670

CONNECTICUT

†Intel Corp.
103 Mill Plain Road
Danbury 06811
Tel: (800) 628-8686
FAX: (203) 794-0339

FLORIDA

†Intel Corp.
800 Fairway Drive
Suite 160
Deerfield Beach 33441
Tel: (800) 628-8686
FAX: (305) 421-2444

Intel Corp.
2250 Lucien Way
Suite 100, Room 8
Maitland 32751
Tel: (800) 628-8686
FAX: (407) 660-1283

GEORGIA

†Intel Corp.
20 Technology Parkway
Suite 150
Norcross 30092
Tel: (800) 628-8686
FAX: (404) 605-9762

IDAHO

Intel Corp.
9456 Fairview Ave., Suite C
Boise 83704
Tel: (800) 628-8686
FAX: (208) 377-1052

ILLINOIS

*†Intel Corp.
Woodfield Corp. Center III
300 N. Martingale Road
Suite 400
Schaumburg 60173
Tel: (800) 628-8686
FAX: (708) 706-9762

INDIANA

†Intel Corp.
8910 Purdue Road
Suite 350
Indianapolis 46268
Tel: (800) 628-8686
FAX: (317) 875-8938

MARYLAND

*†Intel Corp.
10010 Junction Dr.
Suite 200
Annapolis Junction 20701
Tel: (800) 628-8686
FAX: (410) 206-3678

MASSACHUSETTS

*†Intel Corp.
Westford Corp. Center
5 Carlisle Road
2nd Floor
Westford 01886
Tel: (800) 628-8686
TWX: 710-343-6333
FAX: (508) 692-7867

MICHIGAN

†Intel Corp.
7071 Orchard Lake Road
Suite 100
West Bloomfield 48322
Tel: (800) 628-8686
FAX: (313) 851-8770

MINNESOTA

†Intel Corp.
3500 W. 80th St.
Suite 360
Bloomington 55431
Tel: (800) 628-8686
TWX: 910-576-2867
FAX: (612) 831-6497

NEW JERSEY

Intel Corp.
2001 Route 46, Suite 310
Parsippany 07054-1315
Tel: (800) 628-8686
FAX: (201) 402-4893

*†Intel Corp.
Lincroft Office Center
125 Half Mile Road
Red Bank 07701
Tel: (800) 628-8686
FAX: (908) 747-0983

NEW YORK

*Intel Corp.
850 Crosskeys Office Park
Fairport 14450
Tel: (800) 628-8686
TWX: 510-253-7391
FAX: (716) 223-2561

†Intel Corp.
300 Westage Business Center
Suite 230
Fishkill 12524
Tel: (800) 628-8686
FAX: (914) 897-3125

*†Intel Corp.
2950 Express Dr., South
Suite 130
Islandia 11722
Tel: (800) 628-8686
TWX: 510-227-6236
FAX: (516) 348-7939

OHIO

*Intel Corp.
56 Milford Dr., Suite 205
Hudson 44236
Tel: (800) 628-8686
FAX: (216) 528-1026

*†Intel Corp.
3401 Park Center Drive
Suite 220
Dayton 45414
Tel: (800) 628-8686
TWX: 810-450-2528
FAX: (513) 890-8658

OKLAHOMA

Intel Corp.
6801 N. Broadway
Suite 115
Oklahoma City 73162
Tel: (800) 628-8686
FAX: (405) 840-9819

OREGON

†Intel Corp.
15254 N.W. Greenbrier Pkwy.
Building B
Beaverton 97006
Tel: (800) 628-8686
TWX: 910-467-8741
FAX: (503) 645-8181

PENNSYLVANIA

*†Intel Corp.
925 Harvest Drive
Suite 200
Blue Bell 19422
Tel: (800) 628-8686
FAX: (215) 641-0785

SOUTH CAROLINA

Intel Corp.
7403 Parklane Rd., Suite 3
Columbia 29223
Tel: (800) 628-8686
FAX: (803) 788-7999

Intel Corp.
100 Executive Center Drive
Suite 109, B183
Greenville 29615
Tel: (800) 628-8686
FAX: (803) 297-3401

TEXAS

†Intel Corp.
8911 N. Capital of Texas Hwy.
Suite 4230
Austin 78759
Tel: (800) 628-8686
FAX: (512) 338-9335

*†Intel Corp.
5000 Quorum Drive
Suite 750
Dallas 75240
Tel: (800) 628-8686

*†Intel Corp.
20515 SH 249
Suite 401
Houston 77070
Tel: (800) 628-8686
TWX: 910-881-2490
FAX: (713) 988-3660

UTAH

†Intel Corp.
428 East 6400 South
Suite 135
Murray 84107
Tel: (800) 628-8686
FAX: (801) 268-1457

WASHINGTON

†Intel Corp.
2800 156th Avenue S.E.
Suite 105
Bellevue 98007
Tel: (800) 628-8686
FAX: (206) 746-4495

WISCONSIN

Intel Corp.
400 N. Executive Dr.
Suite 401
Brookfield 53005
Tel: (800) 628-8686
FAX: (414) 789-2746

CANADA

BRITISH COLUMBIA

Intel Semiconductor of
Canada, Ltd.
999 Canada Place
Suite 404, #11
Vancouver V6C 3E2
Tel: (800) 628-8686
FAX: (604) 844-2813

ONTARIO

†Intel Semiconductor of
Canada, Ltd.
2650 Queensview Drive
Suite 250
Ottawa K2B 8H6
Tel: (800) 628-8686
FAX: (613) 820-5936

†Intel Semiconductor of
Canada, Ltd.
190 Attwell Drive
Suite 500
Rexdale M9W 6H8
Tel: (800) 628-8686
FAX: (416) 675-2438

QUEBEC

†Intel Semiconductor of
Canada, Ltd.
1 Rue Holiday
Suite 320
Tour East
Pt. Claire H9R 5N3
Tel: (800) 628-8686
FAX: 514-694-0064

†Sales and Service Office
*Field Application Location

NORTH AMERICAN DISTRIBUTORS

ALABAMA

Arrow/Schweber Electronics
1015 Henderson Road
Huntsville 35806
Tel: (205) 837-6955
FAX: (205) 721-1581

Hamilton Hallmark
4890 University Square, #1
Huntsville 35816
Tel: (205) 837-8700
FAX: (205) 830-2565

MTI Systems
4950 Corporate Dr., #120
Huntsville 35805
Tel: (205) 830-9526
FAX: (205) 830-9557

Pioneer Technologies Group
4835 University Square, #5
Huntsville 35805
Tel: (205) 837-9300
FAX: (205) 837-9358

Wyle Laboratories
7800 Governers Drive
Tower Building, 2nd Floor
Huntsville 35806
Tel: (205) 830-1119
FAX: (205) 830-1520

ARIZONA

Anthem Electronics
1555 W. 10th Place, #101
Tempe 85281
Tel: (602) 966-6600
FAX: (602) 966-4826

Arrow/Schweber Electronics
2415 W. Erie Drive
Tempe 85282
Tel: (602) 431-0030
FAX: (602) 252-9109

Avnet Computer
1626 S. Edwards Drive
Tempe 85281
Tel: (602) 902-4600
FAX: (602) 902-4640

Hamilton Hallmark
4637 S. 36th Place
Phoenix 85040
Tel: (602) 437-1200
FAX: (602) 437-2348

Wyle Laboratories
4141 E. Raymond
Phoenix 85040
Tel: (602) 437-2088
FAX: (602) 437-2124

CALIFORNIA

Anthem Electronics
9131 Oakdale Ave.
Chatsworth 91311
Tel: (818) 775-1333
FAX: (818) 775-1302

Anthem Electronics
1 Oldfield Drive
Irvine 92718-2809
Tel: (714) 768-4444
FAX: (714) 768-6456

Anthem Electronics
580 Menlo Drive, #8
Rocklin 95677
Tel: (916) 624-9744
FAX: (916) 624-9750

Anthem Electronics
9369 Carroll Park Drive
San Diego 92121
Tel: (619) 453-9005
FAX: (619) 546-7893

Anthem Electronics
1160 Ridder Park Drive
San Jose 95131
Tel: (408) 452-2219
FAX: (408) 441-4504

Arrow Commercial Systems Group
1502 Crocker Avenue
Hayward 94544
Tel: (510) 489-5371
FAX: (510) 489-9393

Arrow Commercial Systems Group
14242 Chambers Road
Tustin 92680
Tel: (714) 544-0200
FAX: (714) 731-8438

Arrow/Schweber Electronics
26707 W. Agoura Road
Calabasas 91302
Tel: (818) 880-9686
FAX: (818) 772-8930

Arrow/Schweber Electronics
48834 Kato Road, Suite 103
Fremont 94538
Tel: (510) 490-9477

Arrow/Schweber Electronics
6 Cromwell #100
Irvine 92718
Tel: (714) 838-5422
FAX: (714) 454-4206

Arrow/Schweber Electronics
9511 Ridgehaven Court
San Diego 92123
Tel: (619) 565-4800
FAX: (619) 279-8062

Arrow/Schweber Electronics
1180 Murphy Avenue
San Jose 95131
Tel: (408) 441-9700
FAX: (408) 453-4810

Avnet Computer
3170 Pullman Street
Costa Mesa 92626
Tel: (714) 641-4150
FAX: (714) 641-4170

Avnet Computer
1361B West 190th Street
Gardena 90248
Tel: (800) 426-7999
FAX: (310) 327-5389

Avnet Computer
755 Sunrise Boulevard, #150
Roseville 95661
Tel: (916) 781-2521
FAX: (916) 781-3819

Avnet Computer
1175 Bordeaux Drive, #A
Sunnyvale 94089
Tel: (408) 743-3454
FAX: (408) 743-3348

Avnet Computer
21150 Califa Street
Woodland Hills 91376
Tel: (818) 594-8301
FAX: (818) 594-8333

Hamilton Hallmark
3170 Pullman Street
Costa Mesa 92626
Tel: (714) 641-4100
FAX: (714) 641-4122

Hamilton Hallmark
1175 Bordeaux Drive, #A
Sunnyvale 94089
Tel: (408) 435-3500
FAX: (408) 745-6679

Hamilton Hallmark
4545 Viewridge Avenue
San Diego 92123
Tel: (619) 571-7540
FAX: (619) 277-6136

Hamilton Hallmark
21150 Califa St.
Woodland Hills 91367
Tel: (818) 594-0404
FAX: (818) 594-8234

Hamilton Hallmark
580 Menlo Drive, #2
Rocklin 95762
Tel: (916) 624-9781
FAX: (916) 961-0922

Pioneer Standard
5850 Canoga Blvd., #400
Woodland Hills 91367
Tel: (818) 883-4640

Pioneer Standard
217 Technology Dr., #110
Irvine 92718
Tel: (714) 753-5090

Pioneer Technologies Group
134 Rio Robles
San Jose 95134
Tel: (408) 954-9100
FAX: (408) 954-9113

Wyle Laboratories
15370 Barranca Pkwy.
Irvine 92713
Tel: (714) 753-9953
FAX: (714) 753-9877

Wyle Laboratories
15360 Barranca Pkwy., #200
Irvine 92713
Tel: (714) 753-9953
FAX: (714) 753-9877

Wyle Laboratories
2951 Sunrise Blvd., #175
Rancho Cordova 95742
Tel: (916) 638-5282
FAX: (916) 638-1491

Wyle Laboratories
9525 Chesapeake Drive
San Diego 92123
Tel: (619) 565-9171
FAX: (619) 365-0512

Wyle Laboratories
3000 Bowers Avenue
Santa Clara 95051
Tel: (408) 727-2500
FAX: (408) 727-5896

Wyle Laboratories
17872 Cowan Avenue
Irvine 92714
Tel: (714) 863-9953
FAX: (714) 263-0473

Wyle Laboratories
26010 Mureau Road, #150
Calabasas 91302
Tel: (818) 880-9000
FAX: (818) 880-5510

Zeus Arrow Electronics
6276 San Ignacio Ave., #E
San Jose 95119
Tel: (408) 629-4789
FAX: (408) 629-4792

Zeus Arrow Electronics
22700 Savi Ranch Pkwy.
Yorba Linda 92687-4613
Tel: (714) 921-9000
FAX: (714) 921-2715

COLORADO

Anthem Electronics
373 Inverness Drive South
Englewood 80112
Tel: (303) 790-4500
FAX: (303) 790-4532

Arrow/Schweber Electronics
61 Inverness Dr. East, #105
Englewood 80112
Tel: (303) 799-0258
FAX: (303) 373-5760

Hamilton Hallmark
12503 E. Euclid Drive, #20
Englewood 80111
Tel: (303) 790-1662
FAX: (303) 790-4991

Hamilton Hallmark
710 Wooten Road, #102
Colorado Springs 80915
Tel: (719) 637-0055
FAX: (719) 637-0088

Wyle Laboratories
451 E. 124th Avenue
Thornton 80241
Tel: (303) 457-9953
FAX: (303) 457-4831

CONNECTICUT

Anthem Electronics
61 Mattatuck Heights Road
Waterburg 06705
Tel: (203) 575-1575
FAX: (203) 596-3232

Arrow/Schweber Electronics
12 Beaumont Road
Wallingford 06492
Tel: (203) 265-7741
FAX: (203) 265-7988

Avnet Computer
55 Federal Road, #103
Danbury 06810
Tel: (203) 797-2880
FAX: (203) 791-9050

Hamilton Hallmark
125 Commerce Court, Unit 6
Cheshire 06410
Tel: (203) 271-2844
FAX: (203) 272-1704

Pioneer Standard
2 Trap Falls Road
Shelton 06484
Tel: (203) 929-5600

FLORIDA

Anthem Electronics
598 South Northlake Blvd., #1024
Altamonte Springs 32701
Tel: (813) 797-2900
FAX: (813) 796-4880

Arrow/Schweber Electronics
400 Fairway Drive, #102
Deerfield Beach 33441
Tel: (305) 429-8200
FAX: (305) 428-3991

Arrow/Schweber Electronics
37 Skyline Drive, #3101
Lake Mary 32746
Tel: (407) 333-9300
FAX: (407) 333-9320

Avnet Computer
3343 W. Commercial Boulevard
Bldg. C/D, Suite 107
Ft. Lauderdale 33309
Tel: (305) 730-9110
FAX: (305) 730-0368

Avnet Computer
3247 Tech Drive North
St. Petersburg 33716
Tel: (813) 573-5524
FAX: (813) 572-4324

Hamilton Hallmark
3350 N.W. 53rd St., #105-107
Ft. Lauderdale 33309
Tel: (305) 484-5482
FAX: (305) 484-2995

Hamilton Hallmark
10491 72nd St. North
Largo 34647
Tel: (813) 541-7440
FAX: (813) 544-4394

Hamilton Hallmark
7079 University Boulevard
Winter Park 32792
Tel: (407) 657-3300
FAX: (407) 678-4414

Pioneer Technologies Group
337 Northlake Blvd., #1000
Alta Monte Springs 32701
Tel: (407) 834-9090
FAX: (407) 834-0865

Pioneer Technologies Group
674 S. Military Trail
Deerfield Beach 33442
Tel: (305) 428-8877
FAX: (305) 481-2950

Pioneer Technologies Group
8031-2 Phillips Highway
Jacksonville 32256
Tel: (904) 730-0065

Wyle Laboratories
1000 112 Circle North
St. Petersburg 33716
Tel: (813) 530-3400
FAX: (813) 579-1518

GEORGIA

Arrow Commercial Systems Group
3400 C. Corporate Way
Duluth 30136
Tel: (404) 623-8825
FAX: (404) 623-8802

Arrow/Schweber Electronics
4250 E. Rivergreen Pkwy., #E
Duluth 30136
Tel: (404) 497-1300
FAX: (404) 476-1493

Avnet Computer
3425 Corporate Way, #G
Duluth 30136
Tel: (404) 623-5452
FAX: (404) 476-0125

Hamilton Hallmark
3425 Corporate Way, #G & #A
Duluth 30136
Tel: (404) 623-5475
FAX: (404) 623-5490

Pioneer Technologies Group
4250 C. Rivergreen Parkway
Duluth 30136
Tel: (404) 623-1003
FAX: (404) 623-0665

Wyle Laboratories
6025 The Corners Pkwy., #111
Norcross 30092
Tel: (404) 441-9045
FAX: (404) 441-9086

ILLINOIS

Anthem Electronics
1300 Remington Road, Suite A
Schaumberg 60173
Tel: (708) 884-0200
FAX: (708) 885-0480

Arrow/Schweber Electronics
1140 W. Thorndale Rd.
Itasca 60143
Tel: (708) 250-0500

Avnet Computer
1124 Thorndale Avenue
Bensenville 60106
Tel: (708) 860-8572
FAX: (708) 773-7976

Hamilton Hallmark
1130 Thorndale Avenue
Bensenville 60106
Tel: (708) 860-7780
FAX: (708) 860-8530

MTI Systems
1140 W. Thorndale Avenue
Itasca 60143
Tel: (708) 250-8222
FAX: (708) 250-8275

Pioneer Standard
2171 Executive Dr., #200
Addison 60101
Tel: (708) 495-9680
FAX: (708) 495-9831

Wyle Laboratories
2055 Army Trail Road, #140
Addison 60101
Tel: (800) 853-9953
FAX: (708) 620-1610

INDIANA

Arrow/Schweber Electronics
7108 Lakeview Parkway West Dr.
Indianapolis 46268
Tel: (317) 299-2071
FAX: (317) 299-2379

Avnet Computer
485 Gradle Drive
Carmel 46032
Tel: (317) 575-8029
FAX: (317) 844-4964

Hamilton Hallmark
4275 W. 96th
Indianapolis 46268
Tel: (317) 872-8875
FAX: (317) 876-7165

Pioneer Standard
9350 Priority Way West Dr.
Indianapolis 46250
Tel: (317) 573-0880
FAX: (317) 573-0979

NORTH AMERICAN DISTRIBUTORS (Contd.)

KANSAS

Arrow/Schweber Electronics
9801 Legler Road
Lenexa 66219
Tel: (913) 541-9542
FAX: (913) 541-0328

Avnet Computer
15313 W. 95th Street
Lenexa 61219
Tel: (913) 541-7989
FAX: (913) 541-7904

Hamilton Hallmark
10809 Lakeview Avenue
Lenexa 66215
Tel: (913) 888-4747
FAX: (913) 888-0523

KENTUCKY

Hamilton Hallmark
1847 Mercer Road, #G
Lexington 40511
Tel: (800) 235-6039
FAX: (606) 288-4936

MARYLAND

Anthem Electronics
7168A Columbia Gateway Drive
Columbia 21046
Tel: (410) 995-6640
FAX: (410) 290-9862

Arrow Commercial Systems Group
200 Perry Parkway
Gaithersburg 20877
Tel: (301) 670-1600
FAX: (301) 670-0188

Arrow/Schweber Electronics
9800J Patuxent Woods Dr.
Columbia 21046
Tel: (301) 596-7800
FAX: (301) 995-6201

Avnet Computer
7172 Columbia Gateway Dr., #G
Columbia 21045
Tel: (301) 995-3571
FAX: (301) 995-3515

Hamilton Hallmark
10240 Old Columbia Road
Columbia 21046
Tel: (410) 988-9800
FAX: (410) 381-2036

North Atlantic Industries
Systems Division
7125 River Wood Dr.
Columbia 21046
Tel: (301) 312-5800
FAX: (301) 312-5850

Pioneer Technologies Group
15810 Gaither Road
Gaithersburg 20877
Tel: (301) 921-0660
FAX: (301) 670-6746

Wyle Laboratories
7180 Columbia Gateway Dr.
Columbia 21046
Tel: (410) 312-4844
FAX: (410) 312-4953

MASSACHUSETTS

Anthem Electronics
36 Jonspin Road
Wilmington 01887
Tel: (508) 657-5170
FAX: (508) 657-6008

Arrow/Schweber Electronics
25 Upton Dr.
Wilmington 01887
Tel: (508) 658-0900
FAX: (508) 694-1754

Avnet Computer
10 D Centennial Drive
Peabody 01960
Tel: (508) 532-9886
FAX: (508) 532-9660

Hamilton Hallmark
10 D Centennial Drive
Peabody 01960
Tel: (508) 531-7430
FAX: (508) 532-9802

Pioneer Standard
44 Hartwell Avenue
Lexington 02173
Tel: (617) 861-9200
FAX: (617) 863-1547

Wyle Laboratories
15 Third Avenue
Burlington 01803
Tel: (617) 272-7300
FAX: (617) 272-6809

MICHIGAN

Arrow/Schweber Electronics
19880 Haggerty Road
Livonia 48152
Tel: (800) 231-7902
FAX: (313) 462-2686

Avnet Computer
2876 28th Street, S.W., #5
Grandville 49418
Tel: (616) 531-9607
FAX: (616) 531-0059

Avnet Computer
41650 Garden Brook Rd. #120
Novi 48375
Tel: (313) 347-1820
FAX: (313) 347-4067

Hamilton Hallmark
44191 Plymouth Oaks Blvd., #1300
Plymouth 48170
Tel: (313) 416-5800
FAX: (313) 416-5811

Hamilton Hallmark
41650 Garden Brook Rd., #100
Novi 49418
Tel: (313) 347-4271
FAX: (313) 347-4021

Pioneer Standard
4505 Broadmoor S.E.
Grand Rapids 49512
Tel: (616) 698-1800
FAX: (616) 698-1831

Pioneer Standard
13485 Stamford
Livonia 48150
Tel: (313) 525-1800
FAX: (313) 427-3720

MINNESOTA

Anthem Electronics
7646 Golden Triangle Drive
Eden Prairie 55344
Tel: (612) 944-5454
FAX: (612) 944-3045

Arrow/Schweber Electronics
10100 Viking Drive, #100
Eden Prairie 55344
Tel: (612) 941-5280
FAX: (612) 942-7803

Avnet Computer
10000 West 76th Street
Eden Prairie 55344
Tel: (612) 829-0025
FAX: (612) 944-2781

Hamilton Hallmark
9401 James Ave South, #140
Bloomington 55431
Tel: (612) 881-2600
FAX: (612) 881-9461

Pioneer Standard
7625 Golden Triange Dr., #G
Eden Prairie 55344
Tel: (612) 944-3355
FAX: (612) 944-3794

Wyle Laboratories
1325 E. 79th Street, #1
Bloomington 55425
Tel: (612) 853-2280
FAX: (612) 853-2298

MISSOURI

Arrow/Schweber Electronics
2380 Schuetz Road
St. Louis 63141
Tel: (314) 567-6888
FAX: (314) 567-1164

Avnet Computer
741 Goddard Avenue
Chesterfield 63005
Tel: (314) 537-2725
FAX: (314) 537-4248

Hamilton Hallmark
3783 Rider Trail South
Earth City 63045
Tel: (314) 291-5350
FAX: (314) 291-0362

NEW HAMPSHIRE

Avnet Computer
2 Executive Park Drive
Bedford 03102
Tel: (800) 442-8638
FAX: (603) 624-2402

NEW JERSEY

Anthem Electronics
26 Chapin Road, Unit K
Pine Brook 07058
Tel: (201) 227-7960
FAX: (201) 227-9246

Arrow/Schweber Electronics
4 East Stow Rd., Unit 11
Marlton 08053
Tel: (609) 596-8000
FAX: (609) 596-9632

Arrow/Schweber Electronics
43 Route 46 East
Pine Brook 07058
Tel: (201) 227-7880
FAX: (201) 538-4962

Avnet Computer
1-B Keystone Ave., Bldg. 36
Cherry Hill 08003
Tel: (609) 424-8961
FAX: (609) 751-2502

Hamilton Hallmark
1 Keystone Ave., Bldg. 36
Cherry Hill 08003
Tel: (609) 424-0110
FAX: (609) 751-2552

Hamilton Hallmark
10 Lanidex Plaza West
Parsippani 07054
Tel: (201) 515-5300
FAX: (201) 515-1601

MTI Systems
43 Route 46 East
Pinebrook 07058
Tel: (201) 882-8780
FAX: (201) 539-6430

Pioneer Standard
14-A Madison Rd.
Fairfield 07006
Tel: (201) 575-3510
FAX: (201) 575-3454

Wyle Laboratories
20 Chapin Road, Bldg. 10-13
Pinebrook 07058
Tel: (201) 882-8358
FAX: (201) 882-9109

NEW MEXICO

Alliance Electronics, Inc.
10510 Research Ave.
Albuquerque 87123
Tel: (505) 292-3360
FAX: (505) 275-6392

Avnet Computer
7801 Academy Rd.
Bldg. 1, Suite 204
Albuquerque 87109
Tel: (505) 828-9725
FAX: (505) 828-0360

NEW YORK

Anthem Electronics
47 Mall Drive
Commack 11725
Tel: (516) 864-6600
FAX: (516) 493-2244

Arrow/Schweber Electronics
3375 Brighton Henrietta
Townline Rd.
Rochester 14623
Tel: (716) 427-0300
FAX: (716) 427-0735

Arrow/Schweber Electronics
20 Oser Avenue
Hauppauge 11788
Tel: (516) 231-1000
FAX: (516) 231-1072

Avnet Computer
933 Motor Parkway
Hauppauge 11788
Tel: (516) 434-7443
FAX: (516) 434-7426

Avnet Computer
2060 Townline Rd.
Rochester 14623
Tel: (716) 272-9110
FAX: (716) 272-9685

Hamilton Hallmark
933 Motor Parkway
Hauppauge 11788
Tel: (516) 434-7470
FAX: (516) 434-7491

Hamilton Hallmark
1057 E. Henrietta Road
Rochester 14623
Tel: (716) 475-9130
FAX: (716) 475-9119

Hamilton Hallmark
3075 Veterans Memorial Hwy.
Ronkonkoma 11779
Tel: (516) 737-0600
FAX: (516) 737-0838

MTI Systems
1 Penn Plaza
250 W. 34th Street
New York 10119
Tel: (212) 643-1280
FAX: (212) 643-1288

Pioneer Standard
68 Corporate Drive
Binghamton 13904
Tel: (607) 722-9300
FAX: (607) 722-9562

Pioneer Standard
60 Crossway Park West
Woodbury, Long Island 11797
Tel: (516) 921-8700
FAX: (516) 921-2143

Pioneer Standard
840 Fairport Park
Fairport 14450
Tel: (716) 381-7070
FAX: (716) 381-5955

Zeus Arrow Electronics
100 Midland Avenue
Port Chester 10573
Tel: (914) 937-7400
FAX: (914) 937-2553

NORTH CAROLINA

Arrow/Schweber Electronics
5240 Greensdairy Road
Raleigh 27604
Tel: (919) 876-3132
FAX: (919) 878-9517

Avnet Computer
2725 Millbrook Rd., #123
Raleigh 27604
Tel: (919) 790-1735
FAX: (919) 872-4972

Hamilton Hallmark
5234 Greens Dairy Road
Raleigh 27604
Tel: (919) 878-0819
FAX: (919) 878-8729

Pioneer Technologies Group
2200 Gateway Ctr. Blvd, #215
Morrisville 27560
Tel: (919) 460-1530
FAX: (919) 460-1540

OHIO

Arrow Commercial Systems Group
284 Cramer Creek Court
Dublin 43017
Tel: (614) 889-9347
FAX: (614) 889-9680

Arrow/Schweber Electronics
6573 Cochran Road, #E
Solon 44139
Tel: (216) 248-3990
FAX: (216) 248-1106

Arrow/Schweber Electronics
8200 Washington Village Dr.
Centerville 45458
Tel: (513) 435-5563
FAX: (513) 435-2049

Avnet Computer
7764 Washington Village Dr.
Dayton 45459
Tel: (513) 439-6756
FAX: (513) 439-6719

Avnet Computer
30325 Bainbridge Rd., Bldg. A
Solon 44139
Tel: (216) 349-2505
FAX: (216) 349-1894

Hamilton Hallmark
7760 Washington Village Dr.
Dayton 45459
Tel: (513) 439-6735
FAX: (513) 439-6711

Hamilton Hallmark
5821 Harper Road
Solon 44139
Tel: (216) 498-1100
FAX: (216) 248-4803

Hamilton Hallmark
777 Dearborn Park Lane, #L
Worthington 43085
Tel: (614) 888-3313
FAX: (614) 888-0767

MTI Systems
23404 Commerce Park Rd.
Beachwood 44122
Tel: (216) 464-6688
FAX: (216) 464-3564

Pioneer Standard
4433 Interpoint Boulevard
Dayton 45424
Tel: (513) 236-9900
FAX: (513) 236-8133

Pioneer Standard
4800 E. 131st Street
Cleveland 44105
Tel: (216) 587-3600
FAX: (216) 663-1004

OKLAHOMA

Arrow/Schweber Electronics
12101 E. 51st Street, #106
Tulsa 74146
Tel: (918) 252-7537
FAX: (918) 254-0917

Hamilton Hallmark
5411 S. 125th E. Ave., #305
Tulsa 74146
Tel: (918) 254-6110
FAX: (918) 254-6207

Pioneer Standard
9717 E. 42nd St., #105
Tulsa 74146
Tel: (918) 665-7840
FAX: (918) 665-1891

NORTH AMERICAN DISTRIBUTORS (Contd.)

OREGON

Almac Arrow Electronics
1885 N.W. 169th Place
Beaverton 97006
Tel: (503) 629-8090
FAX: (503) 645-0611

Anthem Electronics
9090 S.W. Gemini Drive
Beaverton 97005
Tel: (503) 643-1114
FAX: (503) 626-7928

Avnet Computer
9750 Southwest Nimbus Ave.
Beaverton 97005
Tel: (503) 627-0900
FAX: (502) 526-6242

Hamilton Hallmark
9750 S.W. Nimbus Ave.
Beaverton 97005
Tel: (503) 526-6200
FAX: (503) 641-5939

Wyle Laboratories
9640 Sunshine Court
Bldg. G, Suite 200
Beaverton 97005
Tel: (503) 643-7900
FAX: (503) 646-5466

PENNSYLVANIA

Anthem Electronics
355 Business Center Dr.
Horsham 19044
Tel: (215) 443-5150
FAX: (215) 675-9875

Avnet Computer
213 Executive Drive, #320
Mars 16046
Tel: (412) 772-1888
FAX: (412) 772-1890

Pioneer Technologies Group
259 Kappa Drive
Pittsburgh 15238
Tel: (412) 782-2300
FAX: (412) 963-8255

Pioneer Technologies Group
500 Enterprise Road
Keith Valley Business Center
Horsham 19044
Tel: (713) 530-4700

Wyle Laboratories
1 Eves Drive, #111
Marlton 08053-3185
Tel: (609) 985-7953
FAX: (609) 985-8757

TEXAS

Anthem Electronics
651 N. Plano Road, #401
Richardson 75081
Tel: (214) 238-7100
FAX: (214) 238-0237

Arrow/Schweber Electronics
11500 Metric Blvd., #160
Austin 78758
Tel: (512) 835-4180
FAX: (512) 832-5921

Arrow/Schweber Electronics
3220 Commander Dr.
Carrollton 75006
Tel: (214) 380-6464
FAX: (214) 248-7208

Arrow/Schweber Electronics
10899 Kinghurst Dr., #100
Houston 77099
Tel: (713) 530-4700

Avnet Computer
4004 Beltline, Suite 200
Dallas 75244
Tel: (214) 308-8181
FAX: (214) 308-8129

Avnet Computer
1235 North Loop West, #525
Houston 77008
Tel: (713) 867-8572
FAX: (713) 861-6851

Hamilton Hallmark
12211 Technology Blvd.
Austin 78727
Tel: (512) 258-8848
FAX: (512) 258-3777

Hamilton Hallmark
11420 Page Mill Road
Dallas 75243
Tel: (214) 553-4300
FAX: (214) 553-4395

Hamilton Hallmark
8000 Westglen
Houston 77063
Tel: (713) 781-6100
FAX: (713) 953-8420

Pioneer Standard
1826-D Kramer Lane
Austin 78758
Tel: (512) 835-4000
FAX: (512) 835-9829

Pioneer Standard
13765 Beta Road
Dallas 75244
Tel: (214) 263-3168
FAX: (214) 490-6419

Pioneer Standard
10530 Rockley Road, #100
Houston 77099
Tel: (713) 495-4700
FAX: (713) 495-5642

Wyle Laboratories
1810 Greenville Avenue
Richardson 75081
Tel: (214) 235-9953
FAX: (214) 644-5064

Wyle Laboratories
4030 West Braker Lane, #330
Austin 78758
Tel: (512) 345-8853
FAX: (512) 345-9330

Wyle Laboratories
11001 South Wilcrest, #100
Houston 77099
Tel: (713) 879-9953
FAX: (713) 879-6540

UTAH

Anthem Electronics
1279 West 2200 South
Salt Lake City 84119
Tel: (801) 973-8555
FAX: (801) 973-8909

Arrow/Schweber Electronics
1946 W. Parkway Blvd.
Salt Lake City 84119
Tel: (801) 973-6913
FAX: (801) 972-0200

Avnet Computer
1100 E. 6600 South, #150
Salt Lake City 84121
Tel: (801) 266-1115
FAX: (801) 266-0362

Hamilton Hallmark
1100 East 6600 South, #120
Salt Lake City 84121
Tel: (801) 266-2022
FAX: (801) 263-0104

Wyle Laboratories
1325 West 2200 South, #E
West Valley 84119
Tel: (801) 974-9953
FAX: (801) 972-2524

WASHINGTON

Almac Arrow Electronics
14360 S.E. Eastgate Way
Bellevue 98007
Tel: (206) 643-9992
FAX: (206) 643-9709

Anthem Electronics
19017 - 120th Ave., N.E. #102
Bothell 98011
Tel: (206) 483-1700
FAX: (206) 486-0571

Avnet Computer
17761 N.E. 78th Place
Redmond 98052
Tel: (206) 867-0160
FAX: (206) 867-0161

Hamilton Hallmark
8630 154th Avenue
Redmond 98052
Tel: (206) 881-6697
FAX: (206) 867-0159

Wyle Laboratories
15385 N.E. 90th Street
Redmond 98052
Tel: (206) 881-1150
FAX: (206) 881-1567

WISCONSIN

Arrow/Schweber Electronics
200 N. Patrick, #100
Brookfield 53045
Tel: (414) 792-0150
FAX: (414) 792-0156

Avnet Computer
20875 Crossroads Circle, #400
Waukesha 53186
Tel: (414) 784-8205
FAX: (414) 784-6006

Hamilton Hallmark
2440 S. 179th Street
New Berlin 53146
Tel: (414) 797-7844
FAX: (414) 797-9259

Pioneer Standard
120 Bishop Way #163
Brookfield 53005
Tel: (414) 784-3480
FAX: (414) 780-3613

Wyle Laboratories
W226 N555 Eastmound Drive
Waukesha 53186
Tel: (414) 521-9333
FAX: (414) 521-9498

ALASKA

Avnet Computer
1400 West Benson Blvd., #400
Anchorage 99503
Tel: (907) 274-9899
FAX: (907) 277-2639

CANADA

ALBERTA

Avnet Computer
2816 21st Street Northeast
Calgary T2E 6Z2
Tel: (403) 291-3284
FAX: (403) 250-1591

Zentronics
6815 8th Street N.E., #100
Calgary T2E 7H
Tel: (403) 295-8838
FAX: (403) 295-8714

BRITISH COLUMBIA

Almac Arrow Electronics
8544 Baxter Place
Burnaby V5A 4T8
Tel: (604) 421-2333
FAX: (604) 421-5030

Hamilton Hallmark
8610 Commerce Court
Burnaby V5A 4N6
Tel: (604) 420-4101
FAX: (604) 420-5376

Zentronics
11400 Bridgeport Rd., #108
Richmond V6X 1T2
Tel: (604) 273-5575
FAX: (604) 273-2413

ONTARIO

Arrow/Schweber Electronics
1093 Meyerside, Unit 2
Mississauga L5T 1M4
Tel: (416) 670-7769
FAX: (416) 670-7781

Arrow/Schweber Electronics
36 Antares Dr., Unit 100
Nepean K2E 7W5
Tel: (613) 226-6903
FAX: (613) 723-2018

Avnet Computer
Canada System Engineering Group
151 Superior Blvd.
Mississuaga L5T 2L1
Tel: (416) 795-3835
FAX: (416) 677-5091

Avnet Computer
190 Colonade Road
Nepean K2E 7J5
Tel: (613) 727-2000
FAX: (613) 226-1184

Hamilton Hallmark
151 Superior Blvd., Unit 1-6
Mississauga L5T 2L1
Tel: (416) 564-6060
FAX: (416) 564-6033

Hamilton Hallmark
190 Colonade Road
Nepean K2E 7J5
Tel: (613) 226-1700
FAX: (613) 226-1184

Zentronics
5600 Keaton Crescent, #1
Mississauga L5R 3S5
Tel: (416) 507-2600
FAX: (416) 507-2831

Zentronics
155 Colonnade Rd., South
#17
Nepean K2E 7K1
Tel: (613) 226-8840
FAX: (613) 226-6352

QUEBEC

Arrow/Schweber Electronics
1100 St. Regis Blvd.
Dorval H9P 2T5
Tel: (514) 421-7411
FAX: (514) 421-7430

Arrow/Schweber Electronics
500 Boul. St.-Jean-Baptiste Ave.
Quebec H2E 5R9
Tel: (418) 871-7500
FAX: (418) 871-6816

Avnet Computer
2795 Reu Halpern
St. Laurent H4S 1P8
Tel: (514) 335-2483
FAX: (514) 335-2481

Hamilton Hallmark
7575 Transcanada Highway
#600
St. Laurent H4T 2V6
Tel: (514) 335-1000
FAX: (514) 335-2481

Zentronics
520 McCaffrey
St. Laurent H4T 1N3
Tel: (514) 737-9700
FAX: (514) 737-5212

EUROPEAN SALES OFFICES

FINLAND

Intel Finland OY
Ruosilantie 2
00390 Helsinki
Tel: (358) 0 544 644
FAX: (358) 0 544 030

FRANCE

Intel Corporation S.A.R.L.
1, Rue Edison-BP 303
78054 St. Quentin-en-Yvelines
Cedex
Tel: (33) (1) 30 57 70 00
FAX: (33) (1) 30 64 60 32

GERMANY

Intel GmbH
Dornacher Strasse 1
85622 Feldkirchen/Muenchen
Tel: (49) 089/90992-0
FAX: (49) 089/9043948

ISRAEL

Intel Semiconductor Ltd.
Atidim Industrial Park-Neve Sharet
P.O. Box 43202
Tel-Aviv 61430
Tel: (972) 03 498080
FAX: (972) 03 491870

ITALY

Intel Corporation Italia S.p.A.
Milanofiori Palazzo E
20094 Assago
Milano
Tel: (39) (2) 575441
FAX: (39) (2) 3498464

NETHERLANDS

Intel Semiconductor B.V.
Postbus 84130
3009 CC Rotterdam
Tel: (31) 10 407 11 11
FAX: (31) 10 455 4688

RUSSIA

Intel Technologies, Inc.
Krementshugskaya 6/7
121357 Moscow
Tel: 007-095-4439785
FAX: 007-095-4459420
TLX: 612092 small su.

SPAIN

Intel Iberia S.A.
Zubaran, 28
28010 Madrid
Tel: (34) (1) 308 2552
FAX: (34) (1) 410 7570

SWEDEN

Intel Sweden A.B.
Dalvagen 24
171 36 Solna
Tel: (46) 8 705 5600
FAX: (46) 8 278085

UNITED KINGDOM

Intel Corporation (U.K.) Ltd.
Pipers Way
Swindon, Wiltshire SN3 1RJ
Tel: (44) (0793) 696000
FAX: (44) (0793) 641440

EUROPEAN DISTRIBUTORS/REPRESENTATIVES

AUSTRIA

†*Elbatex GmbH
Eitnergasse 6
A-1231 Wien
Tel: (43) 1816020
FAX: (43) 181652141

†Spoerle Electronic
Heiligenst. Str. 62
A-1190 Wien
Tel: (43) 1 318 72 700
FAX: (43) 1 369 22 73

BELGIUM

†*Inelco Distribution
Avenue des Croix de Guerre 94
1120 Bruxelles
Tel: (32) 2 244 2811
FAX: (32) 2 216 3304

*Diode Belgium
Keiberg II, Minervastraat, 14/B2
1930 Zaventem
Tel: (32) 2 725 46 60
FAX: (32) 2 725 45 11

DENMARK

*Avnet Nortec A/S
Transformervej 17
DK-2730 Herlev
Tel: (45) 4284 2000
FAX: (45) 4492 1552

†*ITT Multikomponent AS
Naverland 29
DK-2600 Glostrup
Tel: (45) 4245 6645
FAX: (45) 4245 7624

FINLAND

†*OY Fintronic AB
Pyyntitie, 3
02230 Espoo
Tel: (358) 0 887 331
FAX: (358) 0 887 33 343

FRANCE

*Arrow Electronique
73-79 Rue des Solets
Silic 585
94663 Rungis Cedex
Tel: (33) (1) 4978 4978
FAX: (33) (1) 4978 0596

*Avnet
79, rue Pierre Semard
92322 Chatillon
Tel: (33) (1) 4965 2500
FAX: (33) (1) 4965 2769

†Metrologie
Tour d'Asnieres
4, Avenue Laurent Cely
92606 Asnieres Cedex
Tel: (33) (1) 4080 9000
FAX: (33) (1) 4791 0561

*Tekelec
Cite des Bruyeres
5, Rue Carle Vernet-BP 2
92310 Sevres
Tel: (33) (1) 4623 2425
FAX: (33) (1) 4507 2191

GERMANY

*Avnet Electronic 2000
Stahlgruberring 12
81829 Muenchen
Tel: (49) 89 45110-01
FAX: (49) 89 45110129

*Jermyn GmbH
Im Dachsstueck 9
65549 Limburg
Tel: (49) 6431 5080
FAX: (49) 6431 508289

†Metrologie GmbH
Steinerstrasse 15
81369 Muenchen
Tel: (49) 89 724470
FAX: (49) 89 72447111

*Proelectron Vertriebs GmbH
Max-Planck-Strasse 1-3
63303 Dreieich
Tel: (49) 6103 304343
FAX: (49) 6103 304425

†Rein Elektronik GmbH
Loetscher Weg 66
41303 Nettetal
Tel: (49) 2153 7330
FAX: (49) 2153 733513

GREECE

†Ergodata
Aigiroupoleos 2A
176 76 Kalithea
Tel: (30) 1 95 10 922
FAX: (30) 1 95 93 160

*Pouliadis Associates Corp.
Aristotelous St. 3/Sygrou Av. 150
Athens 17671
Tel: (30) 1 924 2072
FAX: (30) 1 924 1066

IRELAND

†*Micro Marketing
Taney Hall
Eglinton Terrace
Dundrum
Dublin 14
Tel: (353) (1) 298 9400
FAX: (353) (1) 298 9828

ISRAEL

†*Eastronics Limited
Rozanis 11
P.O.B. 39300
Tel Baruch
Tel-Aviv 61392
Tel: (972) 3 6458 777
FAX: (972) 3 6458 666

ITALY

*Intesi Div. Della Deutsche
Divisione ITT Industries GmbH
P.I. 06550110156
Milanofiori Palazzo e5
20094 Assago (Milano)
Tel: (39) 2 824701
FAX: (39) 2 8242631

*Lasi Elettronica
P.I. 00839000155
Viale Fulvio Testi, N.280
20126 Milano
Tel: (39) 2 661431
FAX: (39) 2 66101385

†Omnilogic Telcom
Via Lorenteggio 270/A
20152 Milano
Tel: (39) 2 48302640
FAX: (39) 2 43802010

NETHERLANDS

†Datelcom B.V.
Meidoornkade 22
3993 AE Houten
Tel: (31) 3403 57222
FAX: (31) 3403 57220

*Diode Components
Coltbaan 17
3439 NG Nieuwegein
Tel: (31) 3402 9 12 34
FAX: (31) 3402 3 59 24

†*Koning en Hartman
Energieweg 1
2627 AP Delft
Tel: (31) 15 609 906
FAX: (31) 15 619 194

NORWAY

*Avnet Nortec A/S
Postboks 123
N-1364 Hvalstad
Tel: (47) 284 6210
FAX: (47) 284 6545

†Computer System Integration A/S
Postbox 198
N-2013 Skjetten
Tel: (47) 638 45 411
FAX: (47) 638 45 310

PORTUGAL

*ATD Electronica LDA
Edificio Altejo
Rua 3 piso 5-sala 505
Urbanizacao de Matinha
1900 Lisboa
Tel: (351) (1) 858 0191 /2
FAX: (351) (1) 858 7841

†Metrologia Iberica Portugal
Rua Dr. Faria de Vasconcelos 3A
1900 Lisboa
Tel: (351) (1) 847 2202
FAX: (351) (1) 847 2197

SOUTH AFRICA

†*EBE
PO Box 912-1222
Silverton 0127
178 Erasmus Street
Meyerspark
Pretoria 0184
Tel: (27) 12 803 7680-93
FAX: (27) 12 803 8294

SPAIN

*ATD Electronica
Avenue de la Industria, 32, 2B
28100 Alcobendas
Madrid
Tel: (34) (1) 661 6551
FAX: (34) (1) 661 6300

†Metrologia Iberica
Avda. Industria, 32-2
28100 Alcobendas
Madrid
Tel: (34) (1) 661 1142
FAX: (34) (1) 661 5755

SWEDEN

†Avnet Computer AB
Box 184
S-123 23 Farsta
Tel: (46) 8 705 18 00
FAX: (46) 8 735 2373

*Avnet Nortec AB
Box 1830
S-171 27 Solna
Tel: (46) 8705 1800
FAX: (46) 883 6918

*ITT Multikomponent AB
Ankdammsgatan 32
Box 1330
S-171 26 Solna
Tel: (46) 8 830020
FAX: (46) 8 27 13 03

SWITZERLAND

†Elbatex AG
Hardstr. 7
CH-5430 Wettingen
Tel: (41) 56 27 50 00
FAX: (41) 27 19 24

†Fabrimex AG
Kirchenweg 5
CH-8032 Zurich
Tel: (41) 1 386 86 86
FAX: (41) 1 383 23 79

†IMIC Microcomputer
Zurichstrasse
CH-8185 Winkel-Ruti
Tel: (41) (1) 8620055
FAX: (41) (1) 8620266

†*Industrade AG
Hertistrasse 31
CH-8304 Wallisellen
Tel: (41) (1) 8328111
FAX: (41) (1) 8307550

TURKEY

*Empa Electronic
Florya Is Merkezi
Besyol Londra Asfalti
34630 Florya Istanbul
Tel: (90) (1) 599 3050
FAX: (90) (1) 599 3061

UNITED KINGDOM

*Arrow Electronics
St. Martins Business Centre
Cambridge Road
Bedford - MK42 0LF
Tel: (44) 234 270272
FAX: (44) 234 211434

*Avnet EMG Ltd.
Jubilee House
Jubilee Road
Letchworth
Hertsfordshire - SG6 1QH
Tel: (44) 462 488 500
FAX: (44) 462 488 567

*Bytech Components
12a Cedarwood
Chineham Business Park
4 Crockford Lane
Basingstoke
Hants RG12 1RW
Tel: (44) 256 707 107
FAX: (44) 256 707 162

†Bytech Systems
5 The Sterling Centre
Eastern Road
Bracknell
Berks - RG12 2PW
Tel: (44) 344 55 333
FAX: (44) 344 867 270

*Datrontech
42-44 Birchett Road
Aldershot
Hants – GU11 1LU
Tel: (44) 252 313155
FAX: (44) 252 341939

*Jermyn Electronics
Vestry Estate
Otford Road
Sevenoaks
Kent TN14 5EU
Tel: (44) 732 743 743
FAX: (44) 732 451 251

†Metrologie VA
Rapid House
Oxford Road
High Wycombe
Bucks - HP11 2E
Tel: (44) 494 526 271
FAX: (44) 494 421 860

*MMD/Rapid Ltd.
Rapid Silicon
3 Bennet Court
Bennet Road
Reading
Berks - RG2 0QX
Tel: (44) 734 750 697
FAX: (44) 734 313 255

*Components
†Systems

INTERNATIONAL SALES OFFICES

AUSTRALIA

Intel Australia Pty. Ltd.
Unit 13
Allambie Grove Business Park
25 Frenchs Forest Road East
Frenchs Forest, NSW, 2086
Sydney
Tel: 61-2-975-3300
FAX: 61-2-975-3375

Intel Australia Pty. Ltd.
711 High Street
1st Floor
East Kw. Vic., 3102
Melbourne
Tel: 61-3-810-2141
FAX: 61-3-819 7200

BRAZIL

Intel Semicondutores do Brasil
Rua Florida, 1703-2 and CJ.22
CEP 04565-001 Sao Paulo
SP Brazil
Tel: 55-11-530-2296
FAX: 55-11-531-5765

CHINA/HONG KONG

Intel PRC Corporation
Room 517-518
China World Tower
1 Jian Guo Men Wai Avenue
Beijing 100004
Republic of China
Tel: 861-505-0386
FAX: 861-505-0383

Intel Semiconductor Ltd.*
32/F Two Pacific Place
88 Queensway
Central
Hong Kong
Tel: (852) 844-4555
FAX: (852) 868-1989

INDIA

Intel Asia Electronics, Inc.
4/2, Samrah Plaza
St. Mark's Road
Bangalore 560001
Tel: 91-80-215065
FAX: 91-80-215067
TLX: 953-845-2646 INTL IN

JAPAN

Intel Japan K.K.
5-6 Tokodai, Tsukuba-shi
Ibaraki, 300-26
Tel: 0298-47-8511
FAX: 0298-47-8450

Intel Japan K.K.*
Hachioji ON Bldg.
4-7-14 Myojin-machi
Hachioji-shi, Tokyo 192
Tel: 0426-48-8770
FAX: 0426-48-8775

Intel Japan K.K.*
Kawa-asa Bldg.
2-11-5 Shin-Yokohama
Kohoku-ku, Yokohama-shi
Kanagawa, 222
Tel: 045-474-7660
FAX: 045-471-4394

Intel Japan K.K.*
Ryokuchi-Eki Bldg.
2-4-1 Terauchi
Toyonaka-shi, Osaka 560
Tel: 06-863-1091
FAX: 06-863-1084

Intel Japan K.K.
Shinmaru Bldg.
1-5-1 Marunouchi
Chiyoda-ku, Tokyo 100
Tel: 03-3201-3621
FAX: 03-3201-6850

Intel Japan K.K.*
TK Gotanda Bldg. 9F
8-3-6 Nishi Gotanda
Shinagawa, Tokyo 141
Tel: 03-3493-6081
FAX: 03-3493-5951

KOREA

Intel Korea, Ltd.
16th Floor, Life Bldg.
61 Yoido-dong, Youngdeungpo-Ku
Seoul 150-010
Tel: (2) 784-8186
FAX: (2) 784-8096

MEXICO

Intel Tecnologia de Mexico
S.A. de C.V.
Av. Mexico No. 2798-9B, S.H.
44680 Guadalajara, Jal.
Tel: 011-523-640-1259
FAX: 011-523-642-7661

SINGAPORE

Intel Singapore Technology, Ltd.
101 Thomson Road #08-05
United Square
Singapore 1130
Tel: (65) 250-7811
FAX: (65) 250-9256

TAIWAN

Intel Technology Far East Ltd.
Taiwan Branch
8th Floor, No. 205
Bank Tower Bldg.
Tung Hua N. Road
Taipei
Tel: 886-2-5144200
FAX: 886-2-717-2455
886-2-719-6184

INTERNATIONAL DISTRIBUTORS/REPRESENTATIVES

ARGENTINA

Dafsys Consulting S.A.
Chacabuco, 90-6 Piso
1069-Buenos Aires
Tel. & FAX: 54.1334.1871

AUSTRALIA

NJS Electronics Australia
1A/37 Ricketts Road
Mount Waverley, VIC 3149
Tel: 61-3-558-9868
FAX: 61-3-558-9929

NSD-Australia
205 Middleborough Rd.
Box Hill, Victoria 3128
Tel: 03 8900970
FAX: 03 8990819

BRAZIL

Hitech
Luis Carlos Berrini, 801 CJ121
04571, Sao Paulo, SP Brazil
Tel: 5511-536-0355
FAX: 5511-240-2650

Microlinear
Avenida Wilhelm Winter, 345
Distrito Industrial - Jundiai, SP
13213-000
Tel: 5511-732-6111
FAX: 5511-732-2892

CHILE

Sisteco
Vecinal 40 – Las Condes
Santiago
Tel: 562-234-1644
FAX: 562-233-9895

CHINA/HONG KONG

Novel Precision Machinery Co., Ltd.
Room 728 Trade Square
681 Cheung Sha Wan Road
Kowloon, Hong Kong
Tel: (852) 360-8999
TWX: 32032 NVTNL HX
FAX: (852) 725-3695

GUATEMALA

Abinitio
11 Calle 2 – Zona 9
Guatemala City
Tel: 5022-32-4104
FAX: 5022-32-4123

INDIA

Priya International Limited
D-6, II Floor
Devatha Plaza
131/132 Residency Rd.
Bangalore 560 025
Tel: 91-80-214027, 91-80-214395
FAX: 91-80-214105

Priya International Limited
Apeejay House, 4th Floor
130 Apollo Street
Bombay 400 023
Tel: 91-22-2660949, 91-22-2665822

Priya International Limited
Flat No. 8, 10th Floor
Akashdeep Building
Barakhamba Rd.
New Delhi 110 001
Tel: 91-11-3314512, 91-11-3310413
FAX: 91-11-3719107

Priya International Limited
5-J, Century Plaza
560-562 Mount Road, Teynampet
Madras 600 018
Tel: 91-44-451031, 91-44-451597
FAX: 91-44-813549

Priya International Limited
No. 10, II Floor, Minerva House
94 Sarojini Devi Rd.
Secunderabad 500 003
Tel: 91-842-813120, 91-842-813549

Priya International Limited
Lords, III Floor
7/1 Lord Sinha Road
Calcutta 700 071
Tel: 91-33-222378, 91-33-222379
FAX: 91-33-224884

SES Computers & Technologies
Pvt. Ltd.
11/18, SNS Chambers
239 Palace Upper Orchards
Sankey Road, Sadashivanagar
Bangalore 560 080
Tel: 91-812-348481
FAX: 91-812-343685

SES Computers & Technologies
Pvt. Ltd.
Arvind Chambers
194, Andheri-Kurla Road
Andheri (East)
Bombay 400 069
Tel: 91-22-6341584, 91-22-6341667
FAX: 91-22-4937524

SES Computers & Technologies
Pvt. Ltd.
605-A, Ansal Chambers II
No. 6, Bhikaji Camaplace
New Delhi 110 066
Tel: 91-11-6881663
FAX: 91-11-6840471

JAMAICA

MC Systems
10-12 Grenada Crescent
Kingston 5
Tel: (809) 926-0104
FAX: (809) 929-5678

JAPAN

Asahi Electronics Co. Ltd.
KMM Bldg. 2-14-1 Asano
Kokurakita-ku
Kitakyushu-shi 802
Tel: 093-511-6471
FAX: 093-551-7861

Dia Semicon Systems, Inc.
Flower Hill Shinmachi Higashi-kan
1-23 Shinmachi, Setagaya-ku
Tokyo 154
Tel: 03-3439-1600
FAX: 03-3439-1601

Okaya Koki
2-4-18 Sakae
Naka-ku, Nagoya-shi 460
Tel: 052-204-8315
FAX: 052-204-8380

Ryoyo Electro Corp.
Konwa Bldg.
1-12-22 Tsukiji
Chuo-ku, Tokyo 104
Tel: 03-3546-5011
FAX: 03-3546-5044

KOREA

Samsung Electronics
Samsung Main Bldg.
150 Taepyung-Ro-2KA, Chung-Ku
Seoul 100-102
C.P.O. Box 8780
Tel: (822) 751-3680
TWX: KORSST K 27970
FAX: (822) 753-9065

Tong Baek Electronic Co., Ltd.
16-58 Hangang-ro 3-ga
Yongsan-gu, Seoul
Tel: 82-2-715-6623
FAX: 82-2-715-9374

SAUDI ARABIA

AAE Systems, Inc.
642 N. Pastoria Ave.
Sunnyvale, CA 94086
U.S.A.
Tel: (408) 732-1710
FAX: (408) 732-3095
TLX: 494-3405 AAE SYS

SINGAPORE

Electronic Resources Pte, Ltd.
17 Harvey Road
#03-01 Singapore 1336
Tel: (65) 283-0888
TWX: RS 56541 ERS
FAX: (65) 289-5327

SOUTH AFRICA

Electronic Building Elements
178 Erasmus St.
(off Watermeyet St.)
Meyerspark, Pretoria, 0184
Tel: 011-2712-803-7680
FAX: 011-2712-803-8294

TAIWAN

Micro Electronics Corporation
12th Floor, Section 3
285 Nanking East Road
Taipei, R.O.C.
Tel: (886) 2-7198419
FAX: (886) 2-7197916

Acer Sertek Inc.
15th Floor, Section 2
Chien Kuo North Rd.
Taipei 18479 R.O.C.
Tel: 886-2-501-0055
TWX: 23756 SERTEK
FAX: (886) 2-5012521

URUGUAY

Interfase
Blvr. Espana 2094
11200 Montevideo
Tel: 5982-49-4600
FAX: 5982-49-3040

VENEZUELA

Unixel C.A.
4 Transversal de Monte Cristo
Edf. AXXA, Piso 1, of. 1&2
Centro Empresarial Boleita
Caracas
Tel: 582-238-7749
FAX: 582-238-1816

*Field Application Location

NORTH AMERICAN SERVICE OFFICES

COMPUTERVISION

Intel Corporation's North American Preferred Service Provider

Central Dispatch: 1-800-876-SERV (1-800-876-7378)

ALABAMA
Birmingham
Huntsville

ALASKA
Anchorage

ARIZONA
Phoenix*
Tucson

ARKANSAS
Little Rock

CALIFORNIA
Bakersfield
Brea
Carson*
Fresno
Livermore
Mar Del Rey
Ontario*
Orange
Sacramento*
San Diego*
San Francisco*
Santa Clara*
Ventura
Sunnyvale
Walnut Creek*
Woodland Hills*

COLORADO
Colorado Springs
Denver
Englewood*

CONNECTICUT
Glastonbury*

DELAWARE
New Castle

FLORIDA
Ft. Lauderdale
Heathrow
Jacksonville
Melbourne
Pensacola
Tampa
West Palm Beach

GEORGIA
Atlanta*
Savannah
West Robbins

HAWAII
Honolulu

ILLINOIS
Buffalo*
Calumer City
Chicago
Lansing
Oak Brook

INDIANA
Carmel*
Ft. Wayne

KANSAS
Overland Park*
Wichita

KENTUCKY
Lexington
Louisville
Madisonville

LOUISIANA
Baton Rouge
Metarie

MAINE
Brunswick

MARYLAND
Frederick
Linthicum*
Rockville*

MASSACHUSETTS
Boston*
Natick*
Norton*
Springfield

MICHIGAN
Ann Harbor
Benton Harbor
Flint
Grand Rapids*
Leslie
Livonia*
St. Joseph
Troy*

MINNESOTA
Bloomington*
Deluth

MISSOURI
Springfield
St. Louis*

NEVADA
Minden
Las Vegas
Reno

NEW HAMSHIRE
Manchester*

NEW JERSEY
Edison*
Hamton Town*
Parsippany*

NEW MEXICO
Albuquerque

NEW YORK
Albany*
Amherst*
Dewitt*
Fairport*
Farmingdale*
New York City*

NORTH CAROLINA
Brevard
Charlotte
Greensboro
Haveluch
Raleigh
Wilmington

NORTH DAKOTA
Bismark

OHIO
Cincinnati*
Columbus
Dayton
Independence*
Middle Heights*
Toledo*

OREGON
Beaverton*

PENNSYLVANIA
Bala Cynwyd*
Camp Hill*
East Erie
Pittsburgh*
Wayne*

SOUTH CAROLINA
Charleston
Cherry Point
Columbia
Fountain Inn

SOUTH DAKOTA
Sioux Falls

TENNESSEE
Bartlett
Chattanooga
Knoxville
Nashville

TEXAS
Austin
Bay City
Beaumont
Canyon
College Station
Houston*
Irving*
San Antonio
Tyler

UTAH
Salt Lake City*

VIRGINIA
Charlottesville
Glen Allen
Maclean*
Norfolk
Virginia Beach

WASHINGTON
Bellevue*
Olympia
Renton
Richland
Spokane
Verdale

WASHINGTON D.C.*

WEST VIRGINIA
St. Albans

WISCONSIN
Brookfield*
Green Bay
Madison
Wausau

CANADA
Calgary*
Edmonton
Halifax
London*
Montreal*
Ottawa
Toronto*
Vancouver, BC*
Winnipeg
Regina
St. John

CUSTOMER TRAINING CENTERS

ARIZONA

Computervision Customer Education
2401 W. Behrend Dr., Suite 17
Phoenix 85027
Tel: 1-800-234-8806

ILLINOIS

Computervision Customer Education
1 Oakbrook Terrace
Suite 600
Oakbrook 60181
Tel: 1-800-234-8806

MASSACHUSETTS

Computervision Customer Education
11 Oak Park Drive
Bedford 01730
Tel: 1-800-234-8806

SYSTEMS ENGINEERING OFFICES

MINNESOTA

3500 W. 80th Street
Suite 360
Bloomington 55431
Tel: (612) 835-6722

NEW YORK

2950 Expressway Dr., South
Islandia 11722
Tel: (506) 231-3300

*Carry-in locations

OEM Boards, Systems & Software

Intel's complete line of original equipment manufacturer (OEM) microcomputer boards, systems, and software takes advantage of Intel's technology to provide economical, computer-based solutions for OEM applications.

The *OEM Boards, Systems, and Software Handbook* features a broad range of hardware and software products for real-time applications and software development; leading-edge Intel technology designed into unique solutions for OEM markets. Included are application notes and data sheets for Intel OEM boards and platforms including such key products as microcomputer platforms, Multibus boards and systems, and iRMX® real-time operating systems. The data sheets include specifications and features of the products as well as architectural and operational descriptions.

*UNIX is a trademark of UNIX System Laboratories

intel®

Order Number: 280407-008
Printed in USA/194/15K/RRD/JP
OEM Systems Products

Printed on Recycled Material

ISBN 1-55512-201-9

90000

9 781555 122010